STOCHASTIC ALGORITHMS FOR DISTRIBUTED OPTIMIZATION AND MACHINE LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Yi Zhou

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

August 2018

STOCHASTIC ALGORITHMS FOR DISTRIBUTED OPTIMIZATION AND MACHINE LEARNING

Approved by:

Dr. Guanghui Lan, Advisor
H. Milton Stewart School of Industrial and System Engineering
Georgia Institute of Technology

Dr. Arkadi Nemirovski
H. Milton Stewart School of Industrial and System Engineering
Georgia Institute of Technology

Dr. Sebastian Pokutta
H. Milton Stewart School of Industrial and System Engineering
Georgia Institute of Technology

Dr. Huan Xu
H. Milton Stewart School of Industrial and System Engineering
Georgia Institute of Technology

Dr. Le Song
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: June 11, 2018

It does not do to dwell on dreams and forget to live

Albus Dumbledore, Harry Potter and the Sorcerer's Stone

To my dear parents Yongming Zhou and Lin Li

ACKNOWLEDGEMENTS

I have come to know a great number of resourceful and respected people during my Ph.D. studies, without whom I can't image how I obtain my Ph.D. degree. To show my sincere appreciation, I would like to name a few here that have helped and advised me to overcome difficulties, discover potentials and become an independent researcher.

First of all, I would love to express my utmost appreciation to my advisor, Guanghui (George) Lan. Professor Lan has offered me tremendous guidance, support and encouragement throughout my Ph.D. studies. He spent countless invaluable hours and efforts to guide me through painstaking moments in research and help me investigate research problems independently. Without him this thesis and my current accomplishments would not happen. I also want to thank Professor Sebastian Pokutta, who has provided me with great supports and suggestions with respect to my research and career development during the past few years. Special thanks should go to Professors Arkadi Nemirovski, Huan Xu and Le Song for serving as my thesis committee members and giving me invaluable advice and guidance on the completion of this thesis.

Moreover, I would like to thank Professors Jean-Philippe Richard and Yongpei Guan from University of Florida. They offered a lot of encouragement and help on my preparations for academic research.

I also want to thank my friends and fellow graduate students from University of Florida and Georgia Tech. They built up my happiest memories during the past five years. Thanks very much for offering me accompany and unconditional supports over the past years. I sincerely wish them all the best in the future.

Last but not least, I would like to thank my parents, Yongming Zhou and Lin Li, who have always been my support and provide me with love and strength. I also want to thank my grandparents, uncles and aunts for the support and useful life advices to help me finding my career path.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

Algorithms

ADPD asynchronous decentralized primal-dual

AA-SDCS asynchronous accelerated stochastic decentralized communication sliding

DCS decentralized communication sliding method

FOMs first-order methods

GEM gradient extrapolation method

NAG Nesterov's accelerated gradient method

PDG primal-dual gradient methods

RGEM random gradient extrapolation method

RIG randomized incremental gradient methods

RPDG randomized primal-dual gradient method

SA stochastic approximation

SDCS stochastic decentralized communication sliding method

SGD stochastic (sub)gradient descent methods

Problem Constants

M Lipschitz constant for f_i

$\hat{L}$ maximum of L_i's

μ strongly convex modulus

L average of L_i's

L_f Lipschitz constant for ∇f

L_i Lipschitz constant for ∇f_i

Graphs

$\mathcal{L}$ the Laplacian matrix of a graph

$\mathcal{E}$ the set of edges

$\mathcal{G}$ an undirected graph

$\mathcal{N}$ the set of agents

Oracles

$\mathcal{SFO}$ stochastic first-order

SUMMARY

In the big data era, machine learning acts as a powerful tool to help us make predictions and decisions, for example, in products recommendation, disease studies, imaging processing and natural language processing, etc. It has strong ties to the field of optimization, in the way the latter provides methods and theory. While data tends to be collected in a distributed fashion, the standard machine learning models require centralizing the training data on one machine or in a data center, which incurs significant communication cost and puts data privacy at risk. To circumvent such an issue, a variety of distributed machine learning models, i.e., optimization problems defined over different network systems, have been proposed and studied in the literature.

Under the setting of distributed convex optimization, each network agent works collaboratively to minimize (resp. maximize) the total system loss (resp. rewards) formulated as a convex (resp. concave) objective function, which is the average/sum of all local objectives associated with the network agents. Similar to the centralized convex optimization, one crucial criterion to evaluate the designed first-order methods for solving distributed optimization problems is the required number of (sub)gradient computations to obtain a solution of certain accuracy, i.e., the sampling complexity of the designed algorithm. In addition, since the individual network agent is ignorant to the full knowledge about the global problem, i.e., the objective functions and data belonging to other agents, they must perform inter-node communications based on the network topology iteratively to propagate and collect distributed information. Therefore, the number of inter-node communication rounds required by the designed algorithm, i.e., the communication complexity, is another important evaluating criterion. As the classical first-order methods are designed for centralized convex optimization problems, they usually lead to huge communication cost and synchronous delays, which is not affordable in the distributed setting, especially in the large-scale network system. In this thesis, we focus on designing and analyzing

efficient stochastic algorithms for distributed machine learning problems that can achieve best-known communication complexities while maintaining optimal sampling complexities.

The first part of this thesis is devoted to investigate randomized incremental gradient (RIG) methods for solving the finite-sum optimization problems, which is the core problem in distributed optimization. By developing and proving the optimality of the primal-dual gradient (PDG) method, which covers a variant of the well-known Nesterov's accelerated gradient (NAG) method as a special case, we propose a randomized version of the PDG method, namely the randomized primal-dual gradient (RPDG) method. Similar to other RIG methods (e.g., SAG and SVRG), RPDG only involves the computation of one randomly selected component function per iteration. Moreover, by providing a lower complexity bound for the class of RIG methods for finite-sum optimization, we demonstrate the optimality of the RPDG method, and this is the first time such an optimal RIG method has been developed for solving finite-sum optimization problems. In comparison with the accelerated stochastic dual coordinate ascent method, RPDG deals with a wider class of problems and can be applied to the cases when the objective function involves a more complicated composite structure and/or a more general regularization term.

In the second part of this thesis, we consider a distributed topology with a central authority, and study the distributed finite-sum optimization problems defined over such star network system. We propose an optimal randomized incremental gradient method, namely the random gradient extrapolation method (RGEM) and show that it does not require any exact gradient evaluations even at the initial point, but can still achieve the optimal communication and sampling complexities for solving finite-sum optimization problems. To the best of our knowledge, this is the first time that such an optimal RIG method without any exact gradient evaluations has been presented for solving finite-sum optimization in the literature. In fact, without any full gradient computation, RGEM possesses iteration costs as low as pure stochastic gradient descent (SGD) methods, but achieves a much faster and

optimal linear rate of convergence for solving deterministic finite-sum problems. In comparison with the well-known randomized Kaczmarz method [1], which can be viewed as an enhanced version of SGD, but can achieve a linear rate of convergence for solving linear systems, RGEM has a better convergence rate in terms of the dependence on the condition number L/μ. Moreover, we extend RGEM for stochastic finite-sum optimization, i.e., we assume that only noisy first-order information of one randomly selected component function can be accessed via a stochastic first-order ($\mathcal{SFO}$) oracle iteratively. In other words, at each iteration only one randomly selected network agent needs to compute an estimator of its gradient by sampling from its local data using a $\mathcal{SFO}$ oracle instead of performing exact gradient evaluation of its component function f_i. Note that for these problems, it is difficult to compute the exact gradients even at the initial point. It also needs to be pointed out that RGEM is developed based on a novel deterministic algorithmic framework, namely gradient extrapolation method (GEM). The development of GEM was inspired by the observation in Section 2.2.2 of Chapter 2 that the NAG method is a special PDG method where the extrapolation step is performed in the primal space. Such a primal extrapolation step, however, might result in a search point outside the feasible region under the randomized setting in the RPDG method mentioned above. In view of this deficiency of PDG and RPDG, we propose to switch the primal and dual spaces for primal-dual gradient methods, and to perform the extrapolation step in the dual (gradient) space. The resulting new first-order method, i.e., GEM, can be viewed as a dual version of Nesterov's accelerated gradient method, and we show that it can also achieve the optimal rate of convergence for black-box convex optimization.

In the third part of this thesis, we study another distributed topology, called the decentralized network topology, where there is no central authority in the distributed networks. Consider communication is the major bottleneck, we present a class of communication-efficient algorithms for solving (stochastic) decentralized nonsmooth optimization problems. We firstly introduce a new decentralized primal-dual type method, called decen-

tralized communication sliding (DCS), where the agents can skip communications while solving their local subproblems iteratively through successive linearizations of their local objective functions. And we show that DCS achieves the best-known complexity bounds on inter-node communication rounds and not improvable sampling complexities. In fact, the sampling complexities are actually comparable to those optimal complexity bounds required for centralized nonsmooth optimization under certain conditions on the target accuracy, and hence are not improvable in general. We also demonstrate that a stochastic version of the DCS method, denoted by SDCS, can achieve the same order of convergence rates as those of DCS on the total number of required communication rounds and stochastic subgradient evaluations. Preliminary numerical experiments are performed to show that DCS and SDCS can significantly save communication costs over some existing state-of-the-art decentralized methods in all our tested instances. Consider synchronization is another critical issue in decentralized optimization other than communication, we further extend the communication-sliding idea to the asynchronous setting. By randomly activating a subset of network agents per iteration, the proposed asynchronous decentralized primal-dual type methods can maintain the communication and sampling complexities obtained by SDCS, but these methods can be applied to solve a broader class of decentralized stochastic problems, for example, composite objective functions with a smooth structure.

CHAPTER 1

INTRODUCTION

In this chapter, we discuss the motivations and background literatures for our research. In particular, we introduce optimization problems defined over a distributed multiagent network connected to a central server, also referred as federated learning [2], as well as reviewing some classic first-order methods (FOMs), for example, stochastic gradient decent (SGD) and randomized incremental gradient (RIG) methods, that can be applied to solve federated learning problems in Section 1.1. We then study decentralized optimization problems defined in the distributed network setting without a central authority and discuss existing decentralized algorithms for decentralized optimization in Section 1.2.

1.1 Distributed Optimization under the Star Topology: Federated Learning

In Section 1.1.1, we study the finite-sum optimization problem defined over the star network topology, where m agents are connected to one central server (or central authority) and all agents only communicate with the server. We then present several well-known first-order methods for solving the finite-sum optimization problems and review their established complexity results in Section 1.1.2 and 1.1.3.

1.1.1 Finite-sum Optimization Problems

The basic problem of interest for federated learning is the finite-sum convex optimization problem given by

$$\Psi^* := \min_{x \in X} \left\{ \Psi(x) := \tfrac{1}{m} \textstyle\sum_{i=1}^{m} f_i(x) + h(x) + \mu\, \omega(x) \right\}. \tag{1.1.1}$$

Here, $X \subseteq \mathbb{R}^n$ is a closed convex set, h is a relatively simple convex function, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$, associated with network agent i are smooth convex functions with Lipschitz continuous gradient, i.e., $\exists L_i \geq 0$ such that

$$\|\nabla f_i(x_1) - \nabla f_i(x_2)\|_* \leq L_i \|x_1 - x_2\|, \ \ \forall x_1, x_2 \in \mathbb{R}^n, \tag{1.1.2}$$

$\omega : X \to \mathbb{R}$ is a strongly convex function with modulus 1 w.r.t. an arbitrary norm $\|\cdot\|$, i.e.,

$$\langle \omega'(x_1) - \omega'(x_2), x_1 - x_2 \rangle \geq \tfrac{1}{2}\|x_1 - x_2\|^2, \ \ \forall x_1, x_2 \in X, \tag{1.1.3}$$

and $\mu \geq 0$ is a given constant. Hence, the objective function Ψ is strongly convex whenever $\mu > 0$. For notational convenience, we also denote $f(x) \equiv \frac{1}{m}\sum_{i=1}^m f_i(x)$, $L \equiv \frac{1}{m}\sum_{i=1}^m L_i$, and $\hat{L} = \max_{i=1,\ldots,m} L_i$. It is easy to see that for some $L_f \geq 0$,

$$\|\nabla f(x_1) - \nabla f(x_2)\|_* \leq L_f \|x_1 - x_2\| \leq L\|x_1 - x_2\|, \ \ \forall x_1, x_2 \in \mathbb{R}^n. \tag{1.1.4}$$

We also consider a class of stochastic finite-sum optimization problems given by

$$\psi^* := \min_{x \in X} \left\{ \psi(x) := \tfrac{1}{m}\textstyle\sum_{i=1}^m \mathbb{E}_{\xi_i}[F_i(x, \xi_i)] + \mu w(x) \right\}, \tag{1.1.5}$$

where ξ_i's are random variables with support $\Xi_i \subseteq \mathbb{R}^d$. It can be easily seen that (1.1.5) is a special case of (1.1.1) with $f_i(x) = \mathbb{E}_{\xi_i}[F_i(x, \xi_i)], \ i = 1, \ldots, m, \ h(\cdot) = 0$. However, different from deterministic finite-sum optimization problems, only noisy gradient information of each component function f_i can be accessed for the stochastic finite-sum optimization problem in (1.1.5).

Throughout this thesis, we assume subproblems of the form

$$\operatorname{argmin}_{x \in X} \langle g, x \rangle + h(x) + \mu\, \omega(x) \tag{1.1.6}$$

are easy to solve for any $g \in \mathbb{R}^n$ and $\mu \geq 0$. We point out below a few examples where such an assumption is satisfied.

- If X is relatively simple, e.g., Euclidean ball, simplex or l_1 ball, and $h(x) = 0$, and $w(\cdot)$ is some properly choosing distance generating function, we can obtain closed form solutions of problem (1.1.6). This is the standard setting used in the regular first-order methods [[3], [4]].

- If the problem is unconstrained, i.e.,$X = \mathcal{E}$, and $h(x)$ is relatively simple, we can derive closed form solutions of (1.1.6) for some interesting cases. For example, if $h(x) = \|x\|_1$ and $w(x) = \|x\|_2^2$, then an explicit solution of (1.1.6) is readily given by its first-order optimality condition. A similar example is given by $h(x) = \sum_{i=1}^d \sigma_i(x)$ and $w(x) = \mathrm{tr}(x^T x)/2$, where $\sigma_i(x)$, $i = 1, \ldots, d$, denote the singular values of $x \in \mathbb{R}^{d \times d}$.

- If X is relatively simple and $h(x)$ is nontrivial, we can still compute closed form solutions of (1.1.6) for some interesting special cases, e.g., when X is the standard simplex, $w(x) = \sum_{i=1}^d x_i \log x_i$ and $h(x) = \sum_{i=1}^d x_i$.

The deterministic finite-sum problem (1.1.1) can model the empirical risk minimization in machine learning and statistical inferences, and hence has become the subject of intensive studies during the past few years. Our study on the finite-sum problems (1.1.1) and (1.1.5) has also been motivated by the emerging need for distributed optimization and machine learning. Under such settings, each component function f_i is associated with an agent i, $i = 1, \ldots, m$, which are connected through a distributed network. While different topologies can be considered for distributed optimization (see, e.g., Figure 1.1 and 1.2), in this section we focus on the star network where m agents are connected to one central server, and all agents only communicate with the server (see Figure 1.1). These types of distributed optimization problems have several unique features. Firstly, they allow for data privacy, since no local data is stored in the server. Secondly, network agents behave inde-

pendently and they may not be responsive at the same time. Thirdly, the communication between the server and agent can be expensive and has high latency. Finally, by considering the stochastic finite-sum optimization problem, we are interested in not only the deterministic empirical risk minimization, but also the generalization risk for distributed machine learning. Moreover, we allow the private data for each agent to be collected in an online (steaming) fashion. One typical example of the aforementioned distributed problems is Federated Learning recently introduced by McMahan et al. in [2]. As a particular example, in the ℓ_2-regularized logistic regression problem, we have

$$f_i(x) = l_i(x) := \frac{1}{N_i}\sum_{j=1}^{N_i} \log(1+\exp(-b_j^i a_j^{i^T} x)),\ i = 1, \ldots, m,\ \ w(x) = R(x) := \frac{1}{2}\|x\|_2^2,$$

provided that f_i is the loss function of agent i with training data $\{a_j^i, b_j^i\}_{j=1}^{N_i} \in \mathbb{R}^n \times \{-1, 1\}$, and $\mu := \lambda$ is the penalty parameter. For minimization of the generalized risk, f_i's are given in the form of expectation, i.e.,

$$f_i(x) = l_i(x) := \mathbb{E}_{\xi_i}[\log(1 + \exp(-\xi_i^T x))],\ i = 1, \ldots, m,$$

where the random variable ξ_i models the underlying distribution for training dataset of agent i. Note that another type of topology for distributed optimization is the multi-agent

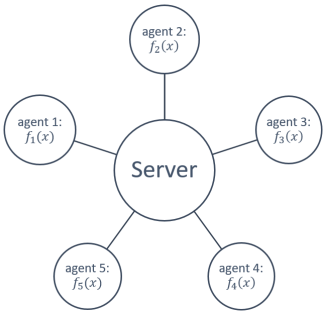

Figure 1.1: A distributed network with 5 agents and one server

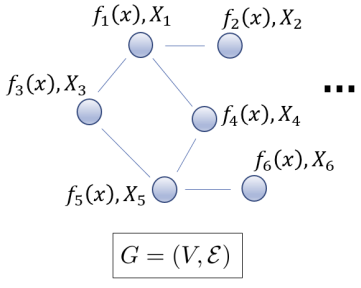

Figure 1.2: An example of the decentralized network

network without a central server, namely the decentralized setting, as shown in Figure 1.2,

where the agents can only communicate with their neighbors to update information, we will discuss this type of distributed problems and the corresponding decentralized algorithms in Section 1.2.

1.1.2 Stochastic Gradient Descent for Finite-sum Optimization

Stochastic (sub)gradient descent (SGD) (a.k.a. stochastic approximation (SA)) type methods have been proven useful to solve problems given in the form of (1.1.1). We will review some classic complexity results of SGD for solving (1.1.1) in this section.

SGD was originally designed to solve stochastic optimization problems given by

$$\min_{x\in X} \mathbb{E}_\xi[F(x,\xi)], \tag{1.1.7}$$

where ξ is a random variable with support $\Xi \subseteq \mathbb{R}^d$. Problem (1.1.1) can be viewed as a special case of (1.1.7) by setting ξ to be a discrete random variable supported on $\{1,\dots,m\}$ with Prob$\{\xi = i\} = \nu_i$ and $F(x,i) = (m\nu_i)^{-1}f_i(x) + h(x) + \mu\omega(x)$, $i = 1,\dots,m$. Since each iteration of SGDs needs to compute the (sub)gradient of only one randomly selected f_i [1], their iteration cost is significantly smaller than that for deterministic first-order methods (FOM), which involves the computation of first-order information of f and thus all the m (sub)gradients of f_i's. Moreover, when f_i's are general nonsmooth convex functions, by properly specifying the probabilities ν_i, $i = 1,\dots,m$ [2], it can be shown (see [3]) that the iteration complexities for both SGD and FOM are in the same order of magnitude. Consequently, the total number of subgradients required by SGDs can be m times smaller than those by FOMs.

Note however, that there is a significant gap on the complexity bounds between SGDs and deterministic FOMs if f_i's are smooth convex functions. For the sake of simplicity,

[1] Observe that the subgradients of h and ω are not required due to the assumption in (1.1.6).

[2] Suppose that f_i are Lipschitz continuous with constants M_i and let us denote $M := \sum_{i=1}^m M_i$, we should set $\nu_i = M_i/M$ in order to get the optimal complexity for SGDs.

let us focus on the strongly convex case when $\mu > 0$ and let x^* be the optimal solution of (1.1.1). In order to find a solution $\bar{x} \in X$ s.t. $\|\bar{x} - x^*\|^2 \leq \epsilon$, the total number of gradient evaluations for f_i's performed by optimal FOMs can be bounded by

$$\mathcal{O}\left\{m\sqrt{\tfrac{L_f}{\mu}}\log\tfrac{1}{\epsilon}\right\}, \tag{1.1.8}$$

which was first achieved by the well-known Nesterov's accelerated gradient method [5, 6], see also relevant extensions in [7, 8, 9]. On the other hand, a direct application of optimal SGDs to the aforementioned stochastic optimization reformulation of (1.1.1) would yield an

$$\mathcal{O}\left\{\sqrt{\tfrac{L}{\mu}}\log\tfrac{1}{\epsilon} + \tfrac{\sigma^2}{\mu\epsilon}\right\} \tag{1.1.9}$$

iteration complexity bound on the number of gradient evaluations for f_i's, which was first achieved by the accelerated stochastic approximation method ([10, 11, 12]). Here $\sigma > 0$ denotes variance of the stochastic gradients, i.e., $\mathbb{E}[\|G(x,\xi) - \nabla f(x)\|_*^2] \leq \sigma^2$, where $G(x,\xi)$ is an unbiased estimator for the gradient $\nabla f(x)$. Clearly, the latter bound is significantly better than the one in (1.1.8) in terms of its dependence on m, but much worse in terms of its dependence on accuracy ϵ and a few other problem parameters (e.g., L and μ). It should be noted that the optimality of (1.1.9) for general stochastic programming (1.1.7) does not preclude the existence of more efficient algorithms for solving (1.1.1), because (1.1.1) is a special case of (1.1.7) with finite support Ξ.

1.1.3 Recent Advancements on Finite-sum Optimization

During the past few years, randomized incremental gradient (RIG) methods, which can access the first-order information of only one randomly selected smooth component f_i at each iteration (see Bertsekas [13] for an introduction to incremental gradient methods), have emerged as an important class of first-order methods for finite-sum optimization (e.g.,[14, 15, 16, 17, 18, 19, 20, 21, 22]). For solving nonsmooth finite-sum problems,

Nemirovski et al. [3, 23] showed that stochastic subgradient (mirror) descent methods can possibly save up to $\mathcal{O}(\sqrt{m})$ subgradient evaluations. By utilizing the smoothness properties of the objective, Lan [24] showed that one can separate the impact of variance from other deterministic components for stochastic gradient descent and presented a new class of accelerated stochastic gradient descent methods to further improve these complexity bounds. However, the overall rate of convergence of these stochastic methods is still sublinear even for smooth and strongly convex finite-sum problems (see [11, 12]). Inspired by these works and the success of the incremental aggregated gradient method by Blatt et al.[14], Schimidt et al. [18] presented a stochastic average gradient (SAG) method, which uses randomized sampling of f_i to update the gradients, and can achieve a linear rate of convergence, i.e., an $\mathcal{O}\{m + (mL/\mu)\log(1/\epsilon)\}$ complexity bound, to solve unconstrained finite-sum problems (3.1.1). Johnson and Zhang later in [15] presented a stochastic variance reduced gradient (SVRG) method, which computes an estimator of ∇f by iteratively updating the gradient of one randomly selected f_i of the current exact gradient information and re-evaluating the exact gradient from time to time. Xiao and Zhang [16] later extended SVRG to solve proximal finite-sum problems (3.1.1). All these methods exhibit an improved $\mathcal{O}\{(m + L/\mu)\log(1/\epsilon)\}$ complexity bound, and Defazio et al. [17] also presented an improved SAG method, called SAGA, that can achieve such a complexity result.

Noting that most of these RIG methods are not optimal even for the deterministic/centralized case (i.e., $m = 1$), much recent research effort has been directed to the acceleration of RIG methods. In Chapter 2, we will proposed a RIG method, namely randomized primal-dual gradient (RPDG) method, and show that its total number of gradient computations of f_i can be bounded by

$$\mathcal{O}\left\{\left(m + \sqrt{\tfrac{mL}{\mu}}\right)\log\tfrac{1}{\epsilon}\right\}. \tag{1.1.10}$$

Evolving from the randomized primal-dual methods developed in [25, 26] for solving saddle-point problems, the RPDG method utilizes a direct acceleration without even using the concept of variance reduction. Simultaneously, Lin et al. [22] presented a catalyst

scheme which utilizes a restarting technique to accelerate the SAG method in [18] (or other "non-accelerated" first-order methods) and thus can possibly improve the complexity bounds obtained by SVRG and SAGA to (1.1.10) (under the Euclidean setting). Allen-Zhu [20] later showed that one can also directly accelerate SVRG to achieve the rate of convergence (1.1.10). All these accelerated RIG methods can save up to $\mathcal{O}(\sqrt{m})$ on the number of gradient evaluations of f_i comparing to optimal deterministic first-order methods when $L/\mu \geq m$. It should be noted that most existing RIG methods were inspired by empirical risk minimization on a single server (or cluster) in machine learning rather than on a set of agents distributed over a network. Under the distributed setting, methods requiring full gradient computation and/or restarting from time to time may incur extra communication and synchronization costs. As a consequence, methods which require fewer full gradient computations (e.g. SAG, SAGA and RPDG) seem to be more advantageous in this regard.

In a related but different line of research, Shalev-Shwartz and Zhang [27] studied a special class of finite-sum optimization problems given in the form of (1.1.1) with $f_i(x)$ given by $\phi_i(a_i^T x)$, where a_i denotes an affine mapping. Under the assumption that $\omega(x) = \|x\|_2^2$, they presented an accelerated stochastic dual coordinate ascent (A-SDCA) method, obtained by properly restarting a stochastic coordinate ascent method in [28] applied to the dual of (1.1.1). Shalev-Shwartz and Zhang show that the iteration complexity of this method can be bounded by (1.1.10). However, each iteration of A-SDCA requires, instead of the computation of ∇f_i, the solution of a subproblem given in the form of

$$\operatorname{argmin}\{\langle g, y\rangle + \phi_i^*(y) + \|y\|_*^2\}, \tag{1.1.11}$$

where ϕ_i^* denotes the conjugate function of ϕ_i. Moreover, these methods were also designed for solving a more special class of problems than (1.1.1). More recently, Lin, Lu, and Xiao [29] proposed to apply the accelerated coordinate descent methods by Nesterov [30], and Fercoq and Richtáriks [31] to obtain similar results for solving these "regularized em-

pirical loss functions" as in [27]. Zhang and Xiao [25] had also obtained similar results by using different stochastic primal-dual coordinate decomposition techniques. Comparing to the class of stochastic dual methods (e.g., [27, 28, 25]), each iteration of the RIG methods only involves the computation ∇f_i, rather than solving a more complicated subproblem (1.1.11) which may not have explicit solutions [27].

An important yet unresolved issue is that there does not exist a valid lower complexity bound for RIG methods in the literature. Hence, it remains unknown what would be the best possible performance that one can expect for these types of methods. Regarding this question, Agarwal and Bottou [32] recently suggested a lower complexity bound for solving problems given in the form of (1.1.1). However, as pointed out by them in a recent ISMP talk in 2015, the lower complexity bound in [32] is deterministic by construction, and hence cannot be used to justify the optimality or suboptimality for the randomized incremental gradient methods in [18, 15, 17] or dual coordinate methods in [29, 27, 25]. In Chapter 2 we will established a lower complexity bound for the RIG methods by showing that the number of gradient evaluations of f_i required by any RIG methods to find an ϵ-solution of (1.1.1), i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[\|\bar{x} - x^*\|_2^2] \leq \epsilon$, cannot be smaller than

$$\Omega\left(\left(m + \sqrt{\tfrac{mL}{\mu}}\right)\log\tfrac{1}{\epsilon}\right), \tag{1.1.12}$$

whenever the dimension n is sufficiently large.

Another interesting but unresolved question in stochastic optimization is whether there exists a method which does not require the computation of any full gradients (even at the initial point), but can still achieve the optimal rate of convergence in (1.1.10). It should be noted that several variants of SAGA, which does not require full gradient computation at the initial point but can still maintain the $\mathcal{O}\{(m + L/\mu)\log(1/\epsilon)\}$ complexity bound, have been recently presented in the literature, see, e.g., [33, 34]. Moreover, little attention in the study of RIG methods has been paid to the stochastic finite-sum problem in (1.1.5), which

is important for generalization risk minimization in machine learning. Very recently, there are some progresses on stochastic primal-dual type methods for solving problem (1.1.5). For example, in Chapter 4 we will proposed a stochastic decentralized communication sliding method that can achieve the optimal $\mathcal{O}(1/\epsilon)$ sampling complexity and best-known $\mathcal{O}(1/\sqrt{\epsilon})$ complexity bounds for communication rounds for solving stochastic decentralized strongly convex problems. For the distributed setting with a central server, by using mini-batch technique to collect gradient information and any stochastic gradient based algorithm as a black box to update iterates, Dekel et al. [35] presented a distributed mini-batch algorithm with a batch size of $o(m^{1/2})$ that can obtain $\mathcal{O}(1/\epsilon)$ sampling complexity (i.e., number of stochastic gradients) for stochastic strongly convex problems, and hence implies at least $\mathcal{O}(1/\sqrt{\epsilon})$ bound for communication complexity. An asynchronous version was later proposed by Feyzmahdavian et al. in [36] that maintained the above convergence rate for regularized stochastic strongly convex problems. It should be pointed out that these mini-batch based distributed algorithms require sampling from all network agents iteratively and hence leads to at least $\mathcal{O}(m/\sqrt{\epsilon})$ rate of convergence in terms of communication costs among server and agents. It is unknown whether there exists an algorithm which only requires a significantly smaller number of communication rounds (e.g., $\mathcal{O}(\log 1/\epsilon)$), but can achieve the optimal $\mathcal{O}(1/\epsilon)$ sampling complexity for solving the stochastic finite-sum problem in (1.1.5).

1.2 Decentralized Optimization

Decentralized optimization problems defined over complex multiagent networks are ubiquitous in signal processing, machine learning, control, and other areas in science and engineering (see e.g. [37, 38, 39, 40]). This section is devoted to discussing the decentralized optimization problems, where in Section 1.2.1 we study the basic problem setup for decentralized optimization and we review the existing decentralized methods and their established complexity results in Section 1.2.2

1.2.1 Problem Setup: Decentralized Problems and the Underlying Network

We consider the following decentralized optimization problem which is cooperatively solved by the network of m agents:

$$\begin{aligned} f^* := \min_x f(x) := \textstyle\sum_{i=1}^m f_i(x) \\ \text{s.t. } x \in X, \quad X := \cap_{i=1}^m X_i, \end{aligned} \tag{1.2.13}$$

where $f_i : X_i \to \mathbb{R}$ is a convex and possibly nonsmooth objective function of agent i satisfying

$$\tfrac{\mu}{2}\|x-y\|^2 \le f_i(x) - f_i(y) - \langle f_i'(y), x-y\rangle \le M\|x-y\|, \ \ \forall x, y \in X_i, \tag{1.2.14}$$

for some $M, \mu \ge 0$ and $f_i'(y) \in \partial f_i(y)$, where $\partial f_i(y)$ denotes the subdifferential of f_i at y, and $X_i \subseteq \mathbb{R}^d$ is a closed convex constraint set of agent i. Note that f_i and X_i are private and only known to agent i. Throughout this thesis, we assume the feasible set X of problem (1.2.13) is nonempty.

We also consider the situation where one can only have access to noisy first-order information (function values and subgradients) of the functions f_i, $i = 1, \ldots, m$ (see [3, 10]). This happens, for example, when the function f_i's are given in the form of expectation, i.e.,

$$f_i(x) := \mathbb{E}_{\xi_i}[F_i(x; \xi_i)], \tag{1.2.15}$$

where the random variable ξ_i models a source of uncertainty and the distribution $\mathbb{P}(\xi_i)$ is not known in advance. As a special case of (1.2.15), f_i may be given as the summation of many components, i.e.,

$$f_i(x) := \textstyle\sum_{j=1}^l f_i^j(x), \tag{1.2.16}$$

where $l \geq 1$ is a large number. Stochastic optimization problem of this type has great potential of applications in data analysis, especially in machine learning. In particular, problem (1.2.15) corresponds to the minimization of generalized risk and is particularly useful for dealing with online (streaming) data distributed over a network, while problem (1.2.16) aims at the collaborative minimization of empirical risk.

Currently the dominant approach to solve (1.2.13) is to collect all agents' private data on a server (or cluster) and to apply centralized machine learning techniques. However, this centralization scheme would require agents to submit their private data to the service provider without much control on how the data will be used, in addition to incurring high setup cost related to the transmission of data to the service provider. Decentralized optimization provides a viable approach to deal with these data privacy related issues. Each network agent i is associated with the local objective function $f_i(x)$ and all agents intend to cooperatively minimize the system objective $f(x)$ as the sum of all local objective f_i's in the absence of full knowledge about the global problem and network structure. A necessary feature in decentralized optimization is, therefore, that the agents must communicate with their neighboring agents to propagate the distributed information to every location in the network.

Consider a multiagent network system whose communication is governed by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = [m]$ indexes the set of agents, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ represents the pairs of communicating agents. If there exists an edge from agent i to j which we denote by (i, j), agent i may send its information to agent j and vice versa. Thus, each agent $i \in \mathcal{N}$ can directly receive (resp., send) information only from (resp., to) the agents in its neighborhood

$$N_i = \{j \in \mathcal{N} \mid (i, j) \in \mathcal{E}\} \cup \{i\}, \tag{1.2.17}$$

where we assume that there always exists a self-loop (i, i) for all agents $i \in \mathcal{N}$. Then,

the associated Laplacian $\mathcal{L} \in \mathbb{R}^{m\times m}$ of $\mathcal{G}$ is $\mathcal{L} := \mathcal{D} - \mathcal{A}$ where D is the diagonal degree matrix, and $A \in \mathbb{R}^{m\times m}$ is the adjacency matrix with the property that $A_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$ and $i \neq j$, i.e.,

$$\mathcal{L}_{ij} = \begin{cases} |N_i| - 1 & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \tag{1.2.18}$$

We consider a reformulation of problem (1.2.13) which is a typical technique in the development of decentralized algorithms. We introduce an individual copy x_i of the decision variable x for each agent $i \in \mathcal{N}$ and impose the constraint $x_i = x_j$ for all pairs $(i, j) \in \mathcal{E}$. The transformed problem can be written compactly by using the Laplacian matrix $\mathcal{L}$:

$$\begin{aligned} &\min_{\mathbf{x}} F(\mathbf{x}) := \textstyle\sum_{i=1}^{m} f_i(x_i) \\ &\text{s.t. } \mathbf{Lx} = \mathbf{0}, \quad x_i \in X_i, \text{ for all } i = 1, \ldots, m, \end{aligned} \tag{1.2.19}$$

where $\mathbf{x} = (x_1, \ldots, x_m) \in X_1 \times \ldots \times X_m$, $F : X_1 \times \ldots \times X_m \to \mathbb{R}$, and $\mathbf{L} = \mathcal{L} \otimes I_d \in \mathbb{R}^{md\times md}$. The constraint $\mathbf{Lx} = \mathbf{0}$ is a compact way of writing $x_i = x_j$ for all agents i and j which are connected by an edge. By construction and Theorem 4.2.12 in [41], $\mathbf{L}$ is symmetric positive semidefinite and its null space coincides with the "agreement" subspace, i.e., $\mathbf{L1} = \mathbf{0}$ and $\mathbf{1}^\top\mathbf{L} = \mathbf{0}$. To ensure each node gets information from every other node, we need the following assumption.

Assumption 1 *The graph $\mathcal{G}$ is connected.*

Under Assumption 1, problem (1.2.13) and (1.2.19) are equivalent. We let Assumption 1 be a blanket assumption for the rest of the this thesis.

1.2.2 First-order Methods for Decentralized Optimization

Decentralized optimization has been extensively studied in recent years due to the emergence of large-scale networks. The seminal work on distributed optimization [42, 43] has been followed by distributed incremental (sub)gradient methods and proximal methods [44, 45, 46, 47], and more recently the incremental aggregated gradient methods and its proximal variants [48, 49, 19]. All of these incremental methods are not fully decentralized in a sense that they require a special star network topology in which the existence of a central authority is necessary for operation. To consider a more general distributed network topology without a central authority, a decentralized subgradient algorithm was first proposed in [50], and further studied in many other literature (see e.g. [51, 52, 53, 54, 55]). These subgradient based methods require each node to compute a local subgradient and followed by the communication with neighboring agents iteratively, and achieve rate of convergence as $\mathcal{O}(1/\epsilon^2)$ to obtain an ϵ-optimal solution, i.e., a point $\hat{x} \in X$, s.t., $\mathbb{E}[f(\hat{x}) - f^*] \leq \epsilon$. While the subgradient computation at each step can be inexpensive, due to the fact that one iteration in decentralized optimization is equivalent to at least one communication round among agents, these methods can incur a significant latency for solving (1.2.13). In fact, CPUs in these days can read and write the memory at over 10 - 100 GB per second whereas communication over TCP/IP is about 100 MB per second. Therefore, the gap between intra-node computation and inter-node communication is about 3 orders of magnitude. The communication start-up cost itself is also not negligible as it usually takes a few milliseconds. Improvements on communication complexity can be obtained when the objective function (1.2.13) is smooth and/or strongly convex (see, e.g., [56, 57, 58, 59]). However, these algorithms do not apply to general nonsmooth and stochastic optimization (cf. (1.2.13)-(1.2.15)) to be studied in Chapter 4.

Besides subgradient based methods, another well-known type of decentralized algorithms relies on dual methods (see e.g., [60, 61, 62, 63, 64, 65]), where at each step for a fixed dual variable, the primal variables are solved to minimize some local Lagrangian

related function, then the dual variables associated with the consistency constraints are updated accordingly. More specifically, the decentralized dual decomposition method proposed in [60] obtained an implicit rate of converge for solving the Lagrangian dual problem of (1.2.13) with bounded communication delays. Furthermore, decentralized alternating direction method of multipliers (ADMM) algorithms (see, e.g., [61, 62, 63, 64]) have received much attention recently. For relatively simple convex functions f_i, the decentralized ADMM proposed in [64] has been shown to require $\mathcal{O}(1/\epsilon)$ communications (see also [66] for the application of mirror-prox method for solving these problems). An improved $\mathcal{O}(\log 1/\epsilon)$ complexity bound on communication rounds can be achieved for decentralized ADMM [62, 63] if stronger assumptions, i.e., smoothness and strong convexity, are imposed on f_i. These dual-based methods have been further studied via proximal-gradients [67, 68, 65]. Although dual type methods usually require fewer numbers of iterations (hence, fewer communication rounds) than the subgradient based methods, the local Lagrangian minimization problem associated with each agent cannot be solved efficiently in many cases, especially when the problem is constrained. Second-order approximation methods [69, 70] have been studied in order to handle this issue, but due to the nature of these methods differentiability of the objective function is necessary in this case.

Moreover, multi-step consensus has been considered in decentralized methods for solving (1.2.13) with smoothness assumption, and hence these methods require an increasing number of communication rounds iteratively. For example, the distributed Nesterov's accelerated gradient method [71] employs multi-consensus in the inner-loop. Although their method requires $\mathcal{O}(1/\sqrt{\epsilon})$ intra-node gradient computations, inter-node communications must increase at a rate of $\mathcal{O}(\log(k))$ as the iteration k increases. Similarly, the proximal gradient method with adapt-then-combine (ATC) multi-consensus strategy and Nesterov's acceleration under the assumption of bounded and Lipschitz gradients [72] requires that inter-node communications must increase at a rate of $\mathcal{O}(k)$. However, the multi-consensus schemes in nested loop algorithms are less desirable, since they do not account for the fact

that the time required for inter-node communications is higher by a few orders of magnitude than that for intra-node computations.

While decentralized algorithms for solving deterministic optimization problems have been extensively studied during the past few years, there exists only limited research on decentralized stochastic optimization, for which only noisy gradient information of functions f_i, $i = 1, \ldots, m$, in (1.2.13) can be easily computed. Existing decentralized stochastic first-order methods for problem (1.2.13) (e.g., [51, 73]) require $\mathcal{O}(1/\epsilon^2)$ inter-node communications and intra-node gradient computations to obtain an ϵ-optimal solution for solving general convex problems. When the objective functions are strongly convex, multiagent mirror descent method for decentralized stochastic optimization can achieve an $\mathcal{O}(1/\epsilon)$ complexity bound [74]. An alternative form of mirror descent in the multiagent setting was proposed by [75] with an asymptotic convergence result. On a broader scale, decentralized stochastic optimization was also considered in the case of time-varying objective functions in the recent work [76, 77]. All these previous works in decentralized stochastic optimization suffered from high communication costs due to the coupled scheme for stochastic subgradient evaluation and communication, i.e., each evaluation of stochastic subgradient will incur one round of communication.

Most of the decentralized algorithms we discussed above are designed under the synchronous setting. However, one critical issue existing in decentralized optimization is that synchrony among network agents is usually inefficient or impractical due to processing and communication delays and the absence of a master server in the network. Note that f_i and X_i are private and only known to agent i, and all agents intend to cooperatively minimize the system objective f as the sum of all local objective f_i's in the absence of full knowledge about the global problem and network structure. Decentralized algorithms, therefore, require agents to communicate with their neighboring agents iteratively to propagate the distributed information in the network. Under the synchronous setting, all agents must wait for the slowest agent and/or slowest communication channel/edge in the network, and a

global coordinator must be presented for synchronization, which can be extremely expensive in the large-scale decentralized network.

Extensive research work has been conducted in recent years to design asynchronous algorithmic schemes for decentralized optimization. Asynchronous gossip-based method under the edge-based random activation setting has been proposed by [78] to solve averaging consensus problems. Later [79] extended this framework for solving (1.2.13) and established almost surely convergence to the optimal solution when f_i is smooth and convex. Most recently, [80] also achieved almost surely convergence by iteratively activating a subset of agents. Besides (sub)gradient based methods, another well-known approach relies on solving the saddle point formulation of (1.2.13) (see Section **??** for the reformulation), where at each iteration a pair of primal and dual variables is updated alternatively. The distributed ADMM (e.g., [81, 64, 82, 83]) has been studied in different asynchronous setting. More specifically, [81, 83] randomly selected and updated a subset of agents iteratively where [81] assuming f_i being simple convex function and [83] establishing almost surely convergence for smooth convex objectives. [64] employed the node-based random activation and achieved the $\mathcal{O}(1/\epsilon)$ rate of convergence when f_i is a simple convex function, and [82] later established the same rate of convergence by activating one agent per iteration. Most recently, [84] proposed an asynchronous parallel primal-dual type method and established almost surely convergence when f_i is smooth and convex.

Asynchronous decentralized algorithms discussed above require the knowledge of exact (sub)gradients (or function values) of f_i, however, this requirement is not realistic when dealing with minimization of generalized risk and online (streaming) data distributed over a network. There exists limited research on asynchronous decentralized stochastic optimization (e.g., [53, 85, 68], for which only noisy gradient information of functions f_i, $i = 1, \ldots, m$, can be easily computed. While asynchronous decentralized stochastic first-order methods [53, 85] established error bounds when f_i is (strongly) convex, [68] achieved $\mathcal{O}(1/\epsilon^2)$ rate of convergence for smooth and convex problems.

1.3 Outline and Main Contributions of the Thesis

This thesis is organized as follows.

In Chapter 2, we focus on developing optimal RIG method for solving (1.1.1). We present a new class of deterministic FOMs, referred to as the primal-dual gradient (PDG) methods, which can achieve the optimal black-box iteration complexity in (1.1.8) for solving (1.1.1), and we are able to show that PDG covers a variant of the well-known Nesterov's accelerated gradient (NAG) method as a special case. We also develop a randomized primal-dual gradient (RPDG) method, which is an optimal RIG method using only one randomly selected component ∇f_i at each iteration. A variant of PDG, this algorithm incorporates an additional dual prediction step before performing the primal descent step. We prove the optimality of RPDG by showing that the number of gradient evaluations required by any RIG methods to find an ϵ-solution of (1.1.1), i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[\|\bar{x} - x^*\|_2^2] \leq \epsilon$, cannot be smaller than (1.1.12) whenever the dimension n is sufficiently large. Moreover, we generalize RPDG for problems which are not necessarily strongly convex (i.e., $\mu = 0$) and/or involve structured nonsmooth terms f_i. We show that for all these cases, RPDG can save $\mathcal{O}(\sqrt{m})$ times gradient computations (up to certain logarithmic factors) in comparison with the corresponding optimal deterministic FOMs at the cost of making $\mathcal{O}(\sqrt{m})$ times more calls to the prox-oracle, which can possibly be reduced by applying mini-batch techniques.

The main goal of Chapter 3 is to introduce new optimal randomized incremental gradient type methods, namely the randomized gradient extrapolation method (RGEM), to solve a much broader class of finite-sum optimization problems. More specifically, RGEM relax the restrictive assumption (1.1.2), i.e, f_i has Lipschitz continuous gradients over the whole $\mathbb{R}^n$ required by RPDG, to f_i having Lipschitz continuous gradients over the feasible set X (see (3.1.2)). Moreover, RGEM does not require any exact gradient evaluations of f, but can still achieve the optimal rate of convergence (1.1.10). We also extend RGEM to

solve stochastic finite-sum problems (1.1.5). Under standard assumptions for centralized stochastic optimization, i.e., the gradient estimators computed by the stochastic first-order ($\mathcal{SFO}$) are unbiased and have bounded variance, RGEM achieves sublinear rate of convergence in terms of the number of stochastic gradient evaluations. By utilizing the mini-batch technique, RGEM can achieve a complexity bound as (1.1.10) in terms of the number of communication rounds, and each round only involves the communication between the server and a randomly selected agent. It needs to be pointed out that RGEM is developed based on a novel algorithmic framework, namely gradient extrapolation method (GEM), we introduce in Chapter 3 for solving black-box convex optimization (i.e., $m = 1$). The development of GEM was inspired by the observation in Chapter 2 (see Section 2.2.2) that the NAG method is a special version of PDG. And GEM can be viewed as a dual version of the NAG method, and we show that it can achieve the optimal rate of convergence for black-box convex optimization.

In Chapter 4, we develop dual based decentralized algorithms for solving (1.2.13) which are communication efficient and have local subproblems approximately solved by each agent through the utilization of (noisy) first-order information of f_i. We firstly introduce a new decentralized primal-dual type method, called decentralized communication sliding (DCS), and show that agents can still find an ϵ-optimal solution in $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication rounds while maintaining the $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) bound on the total number of intra-node subgradient evaluations when the objective functions are general convex (resp., strongly convex). Secondly, we present a stochastic decentralized communication sliding (SDCS) method for solving stochastic optimization problems and show same order rates of convergence as those of DCS on the total number of required communication rounds and stochastic subgradient evaluations. Only requiring the access to one stochastic subgradient iteratively, SDCS provides a communication-efficient way to deal with streaming data and decentralized machine learning. Thirdly, we demonstrate the possible advantages of our proposed methods through preliminary numerical experiments

for solving decentralized support vector machine (SVM) problems with real data sets. For all our test problems, DCS and SDCS can significantly save communication costs over some existing state-of-the-art decentralized methods.

The communication-efficient algorithms in Chapter 4 are designed to solve (1.2.13) under the synchronous setting, and hence each communication and update involves all agents. Inspired by them, we aim to propose an asynchronous decentralized algorithmic framework to solve (1.2.13) under a more general setting but still maintain the complexity bounds achieved in Chapter 4. We first introduce a doubly randomized primal-dual method as the basic asynchronous framework, namely asynchronous decentralized primal-dual (ADPD) method, which randomly activates a subset of agents per iteration, and hence two rounds of communication between the activated agent and its neighboring agents are performed. We then present a new asynchronous stochastic decentralized primal-dual type method under such framework, called asynchronous accelerated stochastic decentralized communication sliding (AA-SDCS) method, for solving decentralized stochastic optimization problems. It should be pointed out that AA-SDCS is a unified algorithm that can be applied to solve a wild range of problems under the general setting, and it maintains the communication and sampling complexities achieved by SDCS. Moreover, the sampling complexities, of AA-SDCS can achieve a better dependence on the Lipschitz constant L when the objective function contains a smooth component than other existing decentralized stochastic first-order methods. We also demonstrate the advantages of the proposed methods through preliminary numerical experiments for solving decentralized support vector machine (SVM) problems with real data sets under a simulated decentralized setting.

CHAPTER 2

AN OPTIMAL RANDOMIZED INCREMENTAL GRADIENT METHOD

2.1 Overview

In this chapter, we consider a class of finite-sum convex optimization problems whose objective function is given by the average of m (≥ 1) smooth components together with some other relatively simple terms. We first introduce a deterministic primal-dual gradient (PDG) method that can achieve the optimal black-box iteration complexity for solving these composite optimization problems using a primal-dual termination criterion. Our major contribution is to develop a randomized primal-dual gradient (RPDG) method, which needs to compute the gradient of only one randomly selected smooth component at each iteration, but can possibly achieve better complexity than PDG in terms of the total number of gradient evaluations. More specifically, we show that the total number of gradient evaluations performed by RPDG can be $\mathcal{O}(\sqrt{m})$ times smaller, both in expectation and with high probability, than those performed by deterministic optimal first-order methods under favorable situations. We also show that the complexity of the RPDG method is not improvable by developing a new lower complexity bound for a general class of randomized methods for solving large-scale finite-sum convex optimization problems.

The rest of this chapter is organized as follows. We first study the deterministic primal-dual method in Section 2.2. Section 2.3 is devoted to the design and analysis of the randomized primal-dual method for the strongly convex case, as well as the development of the lower complexity bound in (1.1.12). In Section 2.4, we generalize the RPDG method to different classes of CP problems that are not necessarily strongly convex. Important technical results and proofs of the main theorems in Sections 2.2 and 2.3 are provided in Section 2.5. Some brief concluding remarks are made in Section 2.6.

2.1.1 Notation and Terminology

We use $\|\cdot\|$ to denote an arbitrary norm in $\mathbb{R}^n$, which is not necessarily associated with the inner product $\langle\cdot,\cdot\rangle$. We also use $\|\cdot\|_*$ to denote the conjugate norm of $\|\cdot\|$. For any convex function h, $\partial h(x)$ is the set of subdifferential at x. Given any $X \subseteq \mathbb{R}^n$, we say a convex function $h : X \to \mathbb{R}$ is Lipschitz continuous if $|h(x) - h(y)| \leq M_h\|x - y\|$ for any $x, y \in X$. We say that a convex function $f : X \to \mathbb{R}$ is smooth if it is differentiable and its gradients are Lipschitz continuous with Lipschitz constant $L > 0$, i.e., $\|\nabla f(y) - \nabla f(x)\|_* \leq L\|y - x\|$ for any $x, y \in X$. For any $p \geq 1$, $\|\cdot\|_p$ denotes the standard p-norm in $\mathbb{R}^n$, i.e.,

$$\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p, \qquad \text{for any } x \in \mathbb{R}^n.$$

For any real number r, $\lceil r \rceil$ and $\lfloor r \rfloor$ denote the nearest integer to r from above and below, respectively. $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively, denote the set of nonnegative and positive real numbers. $\mathcal{N}$ denotes the set of natural numbers $\{1, 2, \ldots\}$.

2.2 An Optimal Primal-dual Gradient Method

Our goal in this section is to present a novel primal-dual gradient (PDG) method for solving (1.1.1), which will also provide a basis for the development of the randomized primal-dual gradient methods in later sections. We establish the optimal convergence of this algorithm in terms of the primal-dual optimality gap under the assumption that the gradient of f is computed at each iteration. We show that PDG generalizes one variant of the well-known Nesterov's accelerated gradient method, and allows a natural game interpretation, and hence that the latter algorithm also admits a similar interpretation.

2.2.1 Preliminaries: Primal and Dual Prox-functions

In this subsection, we discuss both primal and dual prox-functions (proximity control functions) in the primal and dual spaces, respectively.

Recall that the function $\omega : X \to \mathbb{R}$ in (1.1.1) is strongly convex with modulus 1 with respect to $\|\cdot\|$. We can define a primal *prox-function* associated with ω as

$$P(x^0, x) \equiv P_\omega(x^0, x) := \omega(x) - [\omega(x^0) + \langle \omega'(x^0), x - x^0 \rangle], \tag{2.2.1}$$

where $\omega'(x^0) \in \partial\omega(x^0)$ is an arbitrary subgradient of ω at x^0. Clearly, by the strong convexity of ω, we have

$$P(x^0, x) \geq \tfrac{1}{2}\|x - x^0\|^2, \quad \forall x, x^0 \in X. \tag{2.2.2}$$

Note that the prox-function $P(\cdot, \cdot)$ described above generalizes the Bregman's distance in the sense that ω is not necessarily differentiable (see [86, 87, 88, 89] and references therein). Throughout this chapter, we assume that the prox-mapping associated with X, ω, and h, given by

$$\mathcal{M}_X(g, x^0, \eta) \equiv \mathcal{M}_{X,\omega,h}(g, x^0, \eta) := \arg\min_{x \in X} \left\{ \langle g, x \rangle + h(x) + \mu\,\omega(x) + \eta P(x^0, x) \right\}, \tag{2.2.3}$$

is easily computable for any $x^0 \in X$, $g \in \mathbb{R}^n$, $\mu \geq 0$, and $\eta > 0$. Clearly this is equivalent to the assumption that (1.1.6) is easy to solve. Whenever ω is non-differentiable, we need to specify a particular selection of the subgradient ω' before performing the prox-mapping. We assume throughout this chapter that such a selection of ω' is defined recursively as follows. Denote $x^1 \equiv \mathcal{M}_X(g, x^0, \eta)$. By the optimality condition of (2.2.3), we have

$$g + h'(x^1) + (\mu + \eta)\omega'(x^1) - \eta\omega'(x^0) \in \mathcal{N}_X(x^1),$$

where $\mathcal{N}_X(x^1) := \{v \in \mathbb{R}^n : v^T(x - x^1) \leq 0, \forall x \in X\}$ denotes the normal cone of X at x^1. Once such a $\omega'(x^1)$ satisfying the above relation is identified, we will use it as a subgradient when defining $P(x^1, x)$ in the next iteration. Note that such a subgradient

can be identified without additional computational cost as long as x^1 is obtained, since one needs it to check the optimality condition of (2.2.3) when finding x^1.

Now let us consider the dual space $\mathcal{G}$, where the gradients of f reside, and equip it with the conjugate norm $\|\cdot\|_*$. Let $J_f : \mathcal{G} \to \mathbb{R}$ be the conjugate function of f such that

$$f(x) := \max_{g \in \mathcal{G}} \langle x, g \rangle - J_f(g). \tag{2.2.4}$$

It is clear that J_f is strongly convex with modulus $1/L_f$ w.r.t. $\|\cdot\|_*$ (See Chapter E in [90] for details). Therefore, we can define its associated dual prox-functions and dual prox-mappings as

$$D_f(g^0, g) := J_f(g) - [J_f(g^0) + \langle J_f'(g^0), g - g^0 \rangle], \tag{2.2.5}$$

$$\mathcal{M}_{\mathcal{G}}(-\tilde{x}, g^0, \tau) := \arg\min_{g \in \mathcal{G}} \left\{ \langle -\tilde{x}, g \rangle + J_f(g) + \tau D_f(g^0, g) \right\}, \tag{2.2.6}$$

for any $g^0, g \in \mathcal{G}$. Again, D_f may not be uniquely defined since J_f is not necessarily differentiable. Instead of choosing $J_f' \in \partial J_f$ similarly to ω', we can explicitly specify such selections as will be discussed later in this chapter. Observed that (2.2.6) is in the same form as the primal prox-mapping defined in (2.2.3). More specifically, if we let $h(x) = 0,\ \mu = 1$ in (2.2.3), these two prox-mappings all consist of three terms: a linear inner product term, a strongly convex function, and a prox-function generated by the aforementioned strongly convex function.

The following simple result shows that the computation of the dual prox-mapping associated with D_f is equivalent to the computation of ∇f.

Lemma 2.2.1 *Let $\tilde{x} \in X$ and $g^0 \in \mathcal{G}$ be given and $D_f(g^0, g)$ be defined in (2.2.5). For any $\tau > 0$, let us denote $z = [\tilde{x} + \tau J_f'(g^0)]/(1 + \tau)$. Then we have $\nabla f(z) = \mathcal{M}_{\mathcal{G}}(-\tilde{x}, g^0, \tau)$.*

Proof. In view of the definition of D_f in (2.2.5), we have

$$\begin{aligned}\mathcal{M}_{\mathcal{G}}(-\tilde{x}, g^0, \tau) &= \arg\min_{g\in\mathcal{G}} \left\{-\langle \tilde{x} + \tau J_f'(g^0), g\rangle + (1+\tau) J_f(g)\right\} \\ &= \arg\max_{g\in\mathcal{G}} \left\{\langle z, g\rangle - J_f(g)\right\} = \nabla f(z).\end{aligned}$$

■

2.2.2 Primal-dual Gradient Method, Nesterov's Method, and A Game Interpretation

By the definition of J_f in (2.2.4), problem (1.1.1) is equivalent to:

$$\Psi^* := \min_{x\in X} \max_{g\in\mathcal{G}} \left\{\psi(x, g) := h(x) + \mu\,\omega(x) + \langle x, g\rangle - J_f(g)\right\}. \tag{2.2.7}$$

The primal-dual gradient method in Algorithm 1 can be viewed as a game (For further reference regarding game theory, refer to Chapter 14 in [91]). iteratively performed by a primal player (buyer) and a dual player (supplier) for finding the optimal solution (order quantity and product price) of the saddle point problem in (2.2.7). In this game, both the buyer and supplier have access to their local cost $h(x) + \mu\omega(x)$ and $J_f(g)$, respectively, as well as their interactive cost (or revenue) represented by a bilinear function $\langle x, g\rangle$. Our goal is to design an algorithm such that the buyer and supplier can achieve an equilibrium as soon as possible. In the proposed algorithm, the supplier first applies (2.2.8) to predict the demand $\tilde{x}^t$ based on historical information, i.e., x^{t-1} and x^{t-2}. She then determines in (2.2.9) the price g^t in a way to maximize the predicted profit $\langle \tilde{x}^t, g\rangle - J_f(g)$, regularized by the dual prox-function $D_f(g^{t-1}, g)$ with a certain weight $\tau_t \geq 0$. Once after the supplier has made her decision, the buyer then determines his action according to (2.2.10) in order to minimize the cost $h(x)+\mu\omega(x)+\langle x, g\rangle$, regularized by the primal prox-function $P(x^{t-1}, x)$ with a certain weight $\eta_t \geq 0$.

In order to implement the above primal-dual gradient method, it is more convenient to rewrite step (2.2.9) in a form involving the computation of gradient rather than the

Algorithm 1 The primal-dual gradient method

Let $x^0 = x^{-1} \in X$, and the nonnegative parameters $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.
Set $g^0 = \nabla f(x^0)$.
for $t = 1, \ldots, k$ **do**
Update (x^t, g^t) according to

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \tag{2.2.8}$$
$$g^t = \mathcal{M}_{\mathcal{G}}(-\tilde{x}^t, g^{t-1}, \tau_t). \tag{2.2.9}$$
$$x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t). \tag{2.2.10}$$

end for

dual prox-mapping $\mathcal{M}_{\mathcal{G}}$. In order to do so, we shall specify explicitly the selection of the subgradient J_f' in (2.2.9). Denoting $\underline{x}^0 = x^0$, we can easily see from $g^0 = \nabla f(x^0)$ that $\underline{x}^0 \in \partial J_f(g^0)$ (See Chapter E in [90] for reference). Using this relation and letting $J_f'(g^{t-1}) = \underline{x}^{t-1}$ in $D_f(g^{t-1}, g)$ (see (2.2.5)), we then conclude from Lemma 2.2.1 that for any $t \geq 1$, (2.2.9) reduces to

$$\underline{x}^t = (\tilde{x}^t + \tau_t \underline{x}^{t-1})/(1 + \tau_t) \quad \text{and} \quad g^t = \nabla f(\underline{x}^t).$$

With the above selection of the dual prox-function, we can specialize the primal-dual gradient method as follows.

Algorithm 2 A particular implementation of the primal-dual gradient method

Input: Let $x^0 = x^{-1} \in X$, and the nonnegative parameters $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.

Set $\underline{x}^0 = x^0$.

for $t = 1, 2, \ldots, k$ **do**

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \tag{2.2.11}$$

$$\underline{x}^t = \left(\tilde{x}^t + \tau_t \underline{x}^{t-1}\right)/(1 + \tau_t). \tag{2.2.12}$$

$$g^t = \nabla f(\underline{x}^t). \tag{2.2.13}$$

$$x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t). \tag{2.2.14}$$

end for

Observe that one potential problem associated with this scheme is that the search points $\underline{x}^t$ defined in (2.2.11) and (2.2.12), respectively, may fall outside X. As a result, we need to assume f to be differentiable over $\mathbb{R}^n$. However, it can be shown that by properly specifying α_t and τ_t, we can guarantee $\underline{x}^t \in X$ and thus relax such restrictions on the differentiability of f (see (2.2.32) and (2.2.33) below).

The above PDG method is related to the well-known Nesterov's accelerated gradient (AG) method. Let us focus on a simple variant of the AG method that has been extensively studied in the literature (e.g., [6, 9, 10, 11, 12, 92]). Given $(x^{t-1}, \bar{x}^{t-1}) \in X \times X$, this AG algorithm updates $(x^t, \bar{x}^t)$ by

$$\underline{x}^t = (1 - \lambda_t)\bar{x}^{t-1} + \lambda_t x^{t-1}, \tag{2.2.15}$$

$$g^t = \nabla f(\underline{x}^t), \tag{2.2.16}$$

$$x^t = \mathcal{M}_X(g^t, x^{t-1}, \eta_t), \tag{2.2.17}$$

$$\bar{x}^t = (1 - \lambda_t)\bar{x}^{t-1} + \lambda_t x^t, \tag{2.2.18}$$

for some $\lambda_t \in [0,1]$. By (2.2.15) and (2.2.18), we have

$$\begin{aligned}\underline{x}^t &= (1-\lambda_t)[(1-\lambda_{t-1})\bar{x}^{t-2} + \lambda_{t-1}x^{t-1}] + \lambda_t x^{t-1} \\ &= (1-\lambda_t)[\underline{x}^{t-1} - \lambda_{t-1}x^{t-2} + \lambda_{t-1}x^{t-1}] + \lambda_t x^{t-1} \\ &= (1-\lambda_t)\underline{x}^{t-1} + (1-\lambda_t)\lambda_{t-1}(x^{t-1} - x^{t-2}) + \lambda_t x^{t-1}.\end{aligned}$$

Therefore, (2.2.15) is equivalent to (2.2.11) and (2.2.12) with $\tau_t = (1-\lambda_t)/\lambda_t$ and $\alpha_t = \lambda_{t-1}(1-\lambda_t)/\lambda_t$. Moreover, (2.2.16) is identical to (2.2.14)(and (2.2.10)), and (2.2.18) basically defines the output of the AG algorithm as an ergodic mean of the iterates x^t, i.e., the weighted average of all previous iterates $\{x^r\}_{r=0}^t$. We then conclude that the above variant of Nesterov's AG method is a special case of Algorithm 2 (and Algorithm 1). It should be noted, however, different from Nesterov's method, Algorithm 1 is a unified and simpler algorithm for both smooth and strongly convex problems, while Nesterov's method requires another extrapolation step for strongly convex problems. Its flexibility in the specification of parameters will be used later in the development of the RPDG method. Moreover, the presentation of the PDG method helps us to reveal a natural game interpretation out of the intertwined and somehow mysterious updating of the three search sequences in the AG method.

Algorithm 1 is also closely related to Chambolle and Pock's primal-dual method for solving saddle point problems [93, 94], which explains the origin of its name. At the first glance, the PDG method was obtained simply by incorporating the generalized Bregman distance, whose distance generating function is the conjugate of the objective, into primal-dual method proposed in [93]. However, the actual development was much more technically challenging, and requires us to build up a few basic results from the scratch, which are briefly summarized as follows.

- Chambolle and Pock's method in [94] requires the Euclidean distance rather than the Bregman distance when either the primal or dual objective (or both) are strongly

convex. Otherwise, their convergence analysis would not go through (see Section 5.1 and 6 in the original version of [94]). On the other hand, the incorporation of non-Euclidean Bregman distance is crucial for us to establish the connection between the primal and primal-dual methods. We thus need to develop a few new technical results to analyze the PDG method (e.g., Lemma 2.5.16, Proposition 2.5.19, and Theorem 2.2.3), starting from the optimality conditions for the new prox-mapping subproblems to the linear convergence of the PDG method [1].

- We have to deal with non-differentiable Bregman distances since the conjugate of a smooth function is not necessarily differentiable, while all existing works require the differentiability of Bregman distances. Moreover, we need to appropriately specify the selection of the subgradient used in the Bregman distance in order to develop a primal algorithm which only uses gradients rather than any information about the conjugate dual.

- The complexity of all existing primal-dual methods depends on the diameters for both the primal and dual feasible sets. In order to analyze our primal algorithm, we need to establish the relationship between the primal and dual distances (see Lemma 2.5.14) to derive the complexity bounds dependent on the diameter for the primal feasible set only.

2.2.3 Convergence Properties of the Primal-dual Gradient Method

Our goal in this subsection is to show that Algorithm 1 exhibits an optimal rate of convergence for solving problem (1.1.1). It is worth mentioning that our analysis significantly differs from the previous studies on optimal gradient methods and those on primal-dual methods for saddle point problems.

[1] As pointed out by one anonymous reviewer, the authors of [94] had also later mentioned in the published version of their paper the possibility of incorporating more general Bregman distance for strongly convex problems, although no detailed information is provided.

Given a pair of feasible solutions $\bar{z} = (\bar{x}, \bar{g})$ and $z = (x, g)$ of (2.2.7), we define the primal-dual gap function $Q_f(\bar{z}, z)$ by

$$Q_f(\bar{z}, z) := [h(\bar{x}) + \mu\omega(\bar{x}) + \langle \bar{x}, g\rangle - J_f(g)] - [h(x) + \mu\omega(x) + \langle x, \bar{g}\rangle - J_f(\bar{g})]\,. \tag{2.2.19}$$

It can be easily seen that $\bar{z}$ (resp., z) is an optimal solution of (2.2.7) if and only if $Q_f(\bar{z}, z) \leq 0$ for all $z \in X \times \mathcal{G}$ (resp., $Q_f(\bar{z}, z) \geq 0$ for all $\bar{z} \in X \times \mathcal{G}$). In fact, let $z^* = (x^*, g^*)$ be any solution of (2.2.7), by the definition of saddle points, we obtain $\psi(x^*, g) \leq \psi(x^*, g^*) \leq \psi(x, g^*)$, $\forall z \in X \times \mathcal{G}$. Hence $Q_f(\bar{z}, z) = \psi(\bar{x}, g) - \psi(x, \bar{g}) \leq 0$, for all $z \in X \times \mathcal{G}$, if and only if $\bar{z}$ be any saddle point of (2.2.7). Therefore, one can assess the solution quality of $\bar{z}$ by the primal-dual optimality gap:

$$\mathrm{gap}(\bar{z}) := \max_{z \in X \times \mathcal{G}} Q_f(\bar{z}, z). \tag{2.2.20}$$

It should be noted that $\mathrm{gap}(\bar{z})$ may not be well-defined, for example, when X is unbounded and h is not strictly convex. In these cases, we can define a slightly modified primal-dual gap

$$\mathrm{gap}^*(\bar{z}) := \max\left\{Q_f(\bar{z}, z) : x = x^*, g \in \mathcal{G}\right\}, \tag{2.2.21}$$

for an arbitrary optimal solution x^* of (1.1.1). Since J_f is strongly convex, gap^* is well-defined.

The following result establishes some relationship between the primal optimality gap $\Psi(\bar{x}) - \Psi^*$ and the above primal-dual optimality gaps.

Lemma 2.2.2 *Let $\bar{z} = (\bar{x}, \bar{g}) \in X \times \mathcal{G}$ be a given pair of feasible solutions of (2.2.7) and denote $\bar{g}^* = \nabla f(\bar{x})$. Also let $z^* = (x^*, g^*)$ be a pair of optimal solutions of (2.2.7). Then we have*

$$\Psi(\bar{x}) - \Psi(x^*) = Q_f((\bar{x}, g^*), (x^*, \bar{g}^*)) \leq \mathrm{gap}^*(\bar{z}). \tag{2.2.22}$$

If in addition, X is bounded, then

$$\mathrm{gap}^*(\bar{z}) \leq \mathrm{gap}(\bar{z}). \tag{2.2.23}$$

Proof. It follows from the definitions of $\bar{g}^*$, gap^* and the gap function Q_f that

$$\begin{aligned}
\Psi(\bar{x}) - \Psi(x^*) &= Q_f((\bar{x}, g^*), (x^*, \bar{g}^*)) \\
&= [h(\bar{x}) + \mu\omega(\bar{x}) + \max_{g \in \mathcal{G}} \langle \bar{x}, g\rangle - J_f(g)] - [h(x^*) + \mu\omega(x^*) + \langle x^*, g^*\rangle - J_f(g^*)] \\
&\leq [h(\bar{x}) + \mu\omega(\bar{x}) + \max_{g \in \mathcal{G}} \langle \bar{x}, g\rangle - J_f(g)] - [h(x^*) + \mu\omega(x^*) + \langle x^*, \bar{g}\rangle - J_f(\bar{g})] \\
&= \mathrm{gap}^*(\bar{z}).
\end{aligned}$$

Relation (2.2.23) follows directly from the definitions of gap^* and gap. ∎

Theorem 2.2.3 below describes the main convergence properties of the PDG method. More specifically, we provide in Theorem 2.2.3.a) a constant stepsize policy which works for the strongly convex case where $\mu > 0$, and a different parameter setting that works for the non-strongly convex case with $\mu = 0$ in Theorem 2.2.3.b). Note that for the strongly convex case, we estimate the solution quality for the iterates $x^t, t = 1, \ldots, k$, as well as that for their ergodic mean

$$\bar{x}^k = (\textstyle\sum_{t=1}^k \theta_t)^{-1} \sum_{t=1}^k (\theta_t x^t) \tag{2.2.24}$$

for some $\theta_t \geq 0$, while only establishing the error bounds for $\bar{x}^k$ for the non-strongly convex case. We put the proof of Theorem 2.2.3 in Section 2.5 since it shares many basic elements with the convergence analysis of the RPDG method.

Theorem 2.2.3 *Let x^* be an optimal solution of (1.1.1), x^k and $\bar{x}^k$ be defined in (2.2.10) and (2.2.24), respectively.*

a) Suppose that $\mu > 0$ and that $\{\tau_t\}$, $\{\eta_t\}$, $\{\alpha_t\}$ and $\{\theta_t\}$ are set to

$$\tau_t = \sqrt{\tfrac{2L_f}{\mu}}, \quad \eta_t = \sqrt{2L_f\mu}, \quad \alpha_t = \alpha \equiv \tfrac{\sqrt{2L_f/\mu}}{1+\sqrt{2L_f/\mu}}, \quad \textit{and} \quad \theta_t = \tfrac{1}{\alpha^t}, \ \forall t = 1, \ldots, k. \tag{2.2.25}$$

Then,

$$P(x^k, x^*) \leq \tfrac{\mu+L_f}{\mu}\alpha^k P(x^0, x^*), \tag{2.2.26}$$

$$\Psi(\bar{x}^k) - \Psi(x^*) \leq \text{gap}^*(\bar{z}^k) \leq \mu(1-\alpha)^{-1}\left[1 + \tfrac{L_f}{\mu}(2 + \tfrac{L_f}{\mu})\right]\alpha^k P(x^0, x^*), \tag{2.2.27}$$

$$\Psi(\bar{x}^k) - \Psi(x^*) \leq \text{gap}(\bar{z}^k) \leq \mu(1-\alpha)^{-1}\left[1 + \tfrac{L_f}{\mu}(2 + \tfrac{L_f}{\mu})\right]\alpha^k \max_{x\in X} P(x^0, x). \tag{2.2.28}$$

b) Suppose that $\{\tau_t\}$, $\{\eta_t\}$, $\{\alpha_t\}$ and $\{\theta_t\}$ are set to

$$\tau_t = \frac{t-1}{2}, \quad \eta_t = \frac{4L_f}{t}, \quad \alpha_t = \frac{t-1}{t} \quad \textit{and} \quad \theta_t = t, \ \forall t = 1, \ldots, k. \tag{2.2.29}$$

Then,

$$\Psi(\bar{x}^k) - \Psi(x^*) \leq \text{gap}^*(\bar{z}^k) \leq \frac{8L_f}{k(k+1)}P(x^0, x^*), \tag{2.2.30}$$

$$\Psi(\bar{x}^k) - \Psi(x^*) \leq \text{gap}(\bar{z}^k) \leq \frac{8L_f}{k(k+1)}\max_{x\in X} P(x^0, x). \tag{2.2.31}$$

Observe that when the algorithmic parameters are set to (2.2.25), by the definition of $\underline{x}^t$ in (2.2.12) and using an inductive argument, we can easily show that

$$\underline{x}^k = (1-\alpha^2)x^{k-1} + (1-\alpha)\textstyle\sum_{t=1}^{k-2}(\alpha^{k-t}x^t) + \alpha^k x^0. \tag{2.2.32}$$

In other words, $\underline{x}^k$ can be written as a convex combination of $x^0, \ldots, x^{k-1}$ and hence $\underline{x}^k \in$

X for any $k \geq 1$. Similarly, when the algorithmic parameters are set to (2.2.29), we can show by using induction that

$$\underline{x}^k = \tfrac{2(2k-1)}{k(k+1)} x^{k-1} + \tfrac{2}{k(k+1)} \textstyle\sum_{t=1}^{k-2} (t x^t), \tag{2.2.33}$$

which implies $\underline{x}^k \in X$. Therefore, we only need to assume the differentiability of f over X rather than the whole $\mathbb{R}^n$.

In view of the results obtained in Theorem 2.2.3, the primal-dual gradient method is an optimal method for convex optimization. In fact, the rates of convergence in (2.2.27), (2.2.28), (2.2.30) and (2.2.31) associated with the ergodic mean $\bar{z}^k$ have employed the primal-dual optimality gaps $g^*(\bar{z}^k)$ and $g(\bar{z}^k)$ (defined in (2.2.21) and (2.2.20) respectively), which are stronger than the primal optimality gap $\Psi(\bar{x}^k) - \Psi(x^*)$ used in the previous studies for accelerated gradient methods. Moreover, whenever X is bounded, the primal-dual optimality gap $g(\bar{z}^k)$ gives us a computable online accuracy certificates to check the quality of the solution $\bar{z}^k$ (see [95, 11] for some related discussions). Also observe that each iteration of the PDG method requires the computation of ∇f, and hence all the m components ∇f_i. In the next section, we will develop a randomized PDG method that can possibly save the number of gradient evaluations for ∇f_i by utilizing the finite-sum structure of problem (1.1.1).

2.3 Randomized Primal-dual Gradient Methods

In this section, we present a randomized primal-dual gradient (RPDG) method which needs to compute the gradient of only one randomly selected component function f_i at each iteration. We show that RPDG can possibly achieve a better complexity than PDG in terms of the total number of gradient evaluations.

2.3.1 Multi-dual-player Reformulation and the RPDG Algorithm

We start by introducing a different saddle point reformulation of (1.1.1) than (2.2.7). Let $J_i : \mathcal{Y}_i \to \mathbb{R}$ be the conjugate functions of f_i/m and $\mathcal{Y}_i$, $i = 1, \ldots, m$, denote the dual spaces where the gradients of f_i/m reside. For the sake of notational convenience, let us denote $J(y) := \sum_{i=1}^m J_i(y_i)$, $\mathcal{Y} := \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_m$, and $y = (y_1; y_2; \ldots; y_m)$ for any $y_i \in \mathcal{Y}_i$, $i = 1, \ldots, m$. Clearly, we can reformulate problem (1.1.1) equivalently as a saddle point problem:

$$\Psi^* := \min_{x \in X} \left\{ h(x) + \mu\, \omega(x) + \max_{y \in \mathcal{Y}} \{\langle x, Uy\rangle - J(y)\} \right\}, \tag{2.3.1}$$

where $U \in \mathbb{R}^{n \times nm}$ is given by

$$U := [I, I, \ldots, I]\,. \tag{2.3.2}$$

Here I is the identity matrix in $\mathbb{R}^n$. Given a pair of feasible solutions $\bar{z} = (\bar{x}, \bar{y})$ and $z = (x, y)$ of (2.3.1), we define the primal-dual gap function $Q(\bar{z}, z)$ by

$$Q(\bar{z}, z) := [h(\bar{x}) + \mu\omega(\bar{x}) + \langle \bar{x}, Uy\rangle - J(y)] - [h(x) + \mu\omega(x) + \langle x, U\bar{y}\rangle - J(\bar{y})]\,. \tag{2.3.3}$$

It is well-known that $\bar{z} \in Z \equiv X \times \mathcal{Y}$ is an optimal solution of (2.3.1) if and only if $Q(\bar{z}, z) \le 0$ for all $z \in Z$.

Since $J_i, i = 1, \ldots, m$, are strongly convex with modulus $\sigma_i = m/L_i$ w.r.t. $\|\cdot\|_*$, we can define their associated dual prox-functions and dual prox-mappings as

$$D_i(y_i^0, y_i) := J_i(y_i) - [J_i(y_i^0) + \langle J_i'(y_i^0), y_i - y_i^0\rangle], \tag{2.3.4}$$

$$\mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}, y_i^0, \tau) := \arg\min_{y_i \in \mathcal{Y}_i} \left\{ \langle -\tilde{x}, y_i\rangle + J_i(y_i) + \tau D_i(y_i^0, y_i) \right\}, \tag{2.3.5}$$

for any $y_i^0, y_i \in \mathcal{Y}_i$. Accordingly, we define

$$D(\tilde{y}, y) := \sum_{i=1}^m D_i(\tilde{y}_i, y_i). \tag{2.3.6}$$

Again, D_i may not be uniquely defined since J_i are not necessarily differentiable. However, we will discuss how to specify the particular selection of $J_i' \in \partial J_i$ later in this subsection.

We are now ready to describe the randomized primal-dual method, which is obtained by properly modifying the primal-dual gradient method as follows. Firstly, in (2.3.8), we only compute a randomly selected dual prox-mapping $\mathcal{M}_{\mathcal{Y}_i}$ rather than the dual prox-mapping $\mathcal{M}_{\mathcal{G}}$ as in Algorithm 1. Secondly, in addition to the primal prediction step (2.3.7), we add a new dual prediction step (2.3.9), and then use the predicted dual variable $\tilde{y}^t$ for the computation of the new search point x^t in (2.3.10). It can be easily seen that the RPDG method reduces to the PDG method whenever this algorithm is directly applied to (2.2.7) (i.e., $m = 1$, $\mathcal{Y}_1 = \mathcal{G}$, and $J_1 = J_f$) .

Algorithm 3 A randomized primal-dual gradient (RPDG) method

Let $x^0 = x^{-1} \in X$, and the nonnegative parameters $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ be given.
Set $y_i^0 = \frac{1}{m}\nabla f_i(x^0)$, $i = 1, \ldots, m$.
for $t = 1, \ldots, k$ **do**
Choose i_t according to $\mathrm{Prob}\{i_t = i\} = p_i$, $i = 1, \ldots, m$.
Update $z^t = (x^t, y^t)$ according to

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \tag{2.3.7}$$

$$y_i^t = \begin{cases} \mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}^t, y_i^{t-1}, \tau_t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{2.3.8}$$

$$\tilde{y}_i^t = \begin{cases} p_i^{-1}(y_i^t - y_i^{t-1}) + y_i^{t-1}, & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{2.3.9}$$

$$x^t = \mathcal{M}_X(\textstyle\sum_{i=1}^m \tilde{y}_i^t, x^{t-1}, \eta_t). \tag{2.3.10}$$

end for

Similarly to the PDG method, the RPDG method can be viewed as a game iteratively performed by a buyer and m suppliers for finding the solutions (order quantities and prod-

uct prices) of the saddle point problem in (2.3.1). In this game, both the buyer and suppliers have access to their local cost $h(x) + \mu\omega(x)$ and $J_i(y_i)$, respectively, as well as their interactive cost (or revenue) represented by a bilinear function $\langle x, y_i\rangle$. Also, the buyer has to purchase the same amount of products from each supplier (e.g., for fairness). Although there are m suppliers, in each iteration only a randomly chosen supplier can make price changes according to (2.3.8) using the predicted demand $\tilde{x}^t$. In order to understand the buyer's decision in (2.3.10), let us first denote

$$\hat{y}_i^t := \mathcal{M}_{\mathcal{Y}_i}(-\tilde{x}^t, y_i^{t-1}, \tau_t), \quad i = 1, \ldots, m;\ t = 1, \ldots, k. \tag{2.3.11}$$

In other words, $\hat{y}_i^t$, $i = 1, \ldots, m$, denote the prices that all the suppliers can possibly set up at iteration t. Then we can see that

$$\mathbb{E}_t[\tilde{y}_i^t] = \hat{y}_i^t. \tag{2.3.12}$$

Indeed, we have

$$y_i^t = \begin{cases} \hat{y}_i^t, & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{2.3.13}$$

Hence $\mathbb{E}_t[y_i^t] = p_i\hat{y}_i^t + (1 - p_i)y_i^{t-1}$, $i = 1, \ldots, m$. Using this identity in the definition of $\tilde{y}^t$ in (2.3.9), we obtain (2.3.12). Instead of using $\sum_{i=1}^m \hat{y}_i^t$ in determining his order in (2.3.10), the buyer notices that only one supplier has made a change on the price, and thus uses $\sum_{i=1}^m \tilde{y}_i^t$ to predict the case when all the dual players would modify the prices simultaneously.

In order to implement the above RPDG method, we shall explicitly specify the selection of the subgradient J'_{i_t} in the definition of the dual prox-mapping in (2.3.8). Denoting $\underline{x}_i^0 = x^0$, $i = 1, \ldots, m$, we can easily see from $y_i^0 = \frac{1}{m}\nabla f_i(x^0)$ that $\underline{x}_i^0 \in \partial J_i(y_i^0)$, $i = 1, \ldots, m$. Using this relation and letting $J_i'(y_i^{t-1}) = \underline{x}_i^{t-1}$ in the definition of $D_i(y_i^{t-1}, y_i)$ in (2.3.8)

(see (2.3.4)), we then conclude from Lemma 2.2.1 (with $J_f = J_{i_t}$ and $D_f = D_{i_t}$) and (2.3.8) that for any $t \geq 1$,

$$\underline{x}_{i_t}^t = (\tilde{x}^t + \tau_t \underline{x}_{i_t}^{t-1})/(1+\tau_t), \quad \underline{x}_i^t = \underline{x}_i^{t-1}, \ \forall i \neq i_t;$$
$$y_{i_t}^t = \tfrac{1}{m}\nabla f_{i_t}(\underline{x}_{i_t}^t), \quad y_i^t = y_i^{t-1}, \ \forall i \neq i_t.$$

Moreover, observe that the computation of x^t in (2.3.10) requires an involved computation of $\sum_{i=1}^m \tilde{y}_i^t$. In order to save computational time, we suggest to compute this quantity in a recursive manner as follows. Let us denote $g^t \equiv \sum_{i=1}^m y_i^t$. Clearly, in view of the fact that $y_i^t = y_i^{t-1}$, $\forall i \neq i_t$, we have

$$g^t = g^{t-1} + (y_{i_t}^t - y_{i_t}^{t-1}).$$

Also, by the definition of g^t and (2.3.9), we have

$$\begin{aligned}\textstyle\sum_{i=1}^m \tilde{y}_i^t &= \textstyle\sum_{i\neq i_t} y_i^{t-1} + p_{i_t}^{-1}(y_{i_t}^t - y_{i_t}^{t-1}) + y_{i_t}^{t-1} \\ &= \textstyle\sum_{i=1}^m y_i^{t-1} + p_{i_t}^{-1}(y_{i_t}^t - y_{i_t}^{t-1}) \\ &= g^{t-1} + p_{i_t}^{-1}(y_{i_t}^t - y_{i_t}^{t-1}).\end{aligned}$$

Incorporating these two ideas mentioned above, we present an efficient implementation of the RPDG method in Algorithm 4.

Clearly, the RPDG method is an incremental gradient type method since each iteration of this algorithm involves the computation of the gradient ∇f_{i_t} of only one component function. As shown in the following Subsection, such a randomization scheme can lead to significant savings on the total number of gradient evaluations, at the expense of more primal prox-mappings.

It should also be noted that due to the randomness in the RPDG method, we can not guarantee that $\underline{x}_i^t \in X$ for all $i = 1, \ldots, m$, and $t \geq 1$ in general, even though we do

Algorithm 4 An efficient implementation of the RPDG method

Let $x^0 = x^{-1} \in X$, and nonnegative parameters $\{\alpha_t\}$, $\{\tau_t\}$, and $\{\eta_t\}$ be given.
Set $\underline{x}_i^0 = x^0$, $y_i^0 = \frac{1}{m}\nabla f_i(x^0)$, $i = 1, \ldots, m$, and $g^0 = \sum_{i=1}^m y_i^0$.
for $t = 1, \ldots, k$ **do**
Choose i_t according to $\mathrm{Prob}\{i_t = i\} = p_i$, $i = 1, \ldots, m$.
Update $z^t := (x^t, y^t)$ by

$$\tilde{x}^t = \alpha_t(x^{t-1} - x^{t-2}) + x^{t-1}. \tag{2.3.14}$$

$$\underline{x}_i^t = \begin{cases} (1+\tau_t)^{-1}\left(\tilde{x}^t + \tau_t \underline{x}_i^{t-1}\right), & i = i_t, \\ \underline{x}_i^{t-1}, & i \neq i_t. \end{cases} \tag{2.3.15}$$

$$y_i^t = \begin{cases} \frac{1}{m}\nabla f_i(\underline{x}_i^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{2.3.16}$$

$$x^t = \mathcal{M}_X(g^{t-1} + p_{i_t}^{-1}(y_{i_t}^t - y_{i_t}^{t-1}), x^{t-1}, \eta_t). \tag{2.3.17}$$

$$g^t = g^{t-1} + y_{i_t}^t - y_{i_t}^{t-1}. \tag{2.3.18}$$

end for

have all the iterates $x^t \in X$. That is why we need to make the assumption that f_i's are differentiable over $\mathbb{R}^n$ for the RPDG method.

2.3.2 The Convergence of the RPDG Algorithm

Our goal in this subsection is to describe the convergence properties of the RPDG method for the strongly convex case when $\mu > 0$. Generalization of the RPDG method for the non-strongly convex case will be discussed in Section 2.4.

Theorem 2.3.4 below states some general convergence properties of RPDG. Similar to PDG method, we provide bounds on $\mathbb{E}[P(x^k, x^*)]$ and $\mathbb{E}[\Psi(\bar{x}^k) - \Psi(x^*)]$. However, we cannot provide a bound on the expected primal-dual gap $\mathbb{E}[\mathrm{gap}(\bar{x}^k)]$ even though our analysis for the RPDG algorithm still relies on the primal-dual gap function Q in (2.3.3) (see [26] for some relevant discussions).

Theorem 2.3.4 *Suppose that $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ in the RPDG method are set to*

$$\tau_t = \tau, \quad \eta_t = \eta, \quad \textit{and} \quad \alpha_t = \alpha, \tag{2.3.19}$$

for any $t \geq 1$ such that

$$(1-\alpha)(1+\tau) \leq p_i, i = 1, \ldots, m, \tag{2.3.20}$$

$$\eta \leq \alpha(\mu + \eta), \tag{2.3.21}$$

$$\eta\tau p_i \geq 4L_i/m, i = 1, \ldots, m, \tag{2.3.22}$$

for some $\alpha \in (0, 1)$. Then, for any $k \geq 1$, we have

$$\mathbb{E}[P(x^k, x^*)] \leq \left(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta}\right)\alpha^k P(x^0, x^*), \tag{2.3.23}$$

$$\mathbb{E}[\Psi(\bar{x}^k) - \Psi(x^*)] \leq \alpha^{k/2}\left(\alpha^{-1}\eta + \tfrac{3-2\alpha}{1-\alpha}L_f + \tfrac{2L_f^2\alpha}{(1-\alpha)\eta}\right)P(x^0, x^*), \tag{2.3.24}$$

where $\bar{x}^k = (\sum_{t=1}^k \theta_t)^{-1}\sum_{t=1}^k(\theta_t x^t)$ with $\{\theta_t\}$ defined as in (2.2.25), and x^* denotes the optimal solution of problem (1.1.1), and the expectation is taken w.r.t. $i_1, \ldots, i_k$.

We now provide a few specific selections of p_i, τ, η, and α satisfying (2.3.20)-(2.3.22) and establish the complexity of the RPDG method for computing a stochastic ϵ-solution of problem (1.1.1), i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[P(\bar{x}, x^*)] \leq \epsilon$, as well as a stochastic (ϵ, λ)-solution of problem (1.1.1), i.e., a point $\bar{x} \in X$ s.t. Prob$\{P(\bar{x}, x^*) \leq \epsilon\} \geq 1 - \lambda$ for some $\lambda \in (0, 1)$. Moreover, in view of (2.3.24), similar complexity bounds of the RPDG method can be established in terms of the primal optimality gap, i.e. $\mathbb{E}[\Psi(\bar{x}) - \Psi^*]$.

The following corollary shows the convergence of RPDG under a non-uniform distribution for the random variables i_t, $t = 1, \ldots, k$.

Corollary 2.3.5 *Suppose that $\{i_t\}$ in the RPDG method are distributed over $\{1, \ldots, m\}$ according to*

$$p_i = \text{Prob}\{i_t = i\} = \tfrac{1}{2m} + \tfrac{L_i}{2mL}, i = 1, \ldots, m. \tag{2.3.25}$$

Also assume that $\{\tau_t\}$, $\{\eta_t\}$, *and* $\{\alpha_t\}$ *are set to (2.3.19) with*

$$\tau = \frac{\sqrt{(m-1)^2+4mC}-(m-1)}{2m},\ \ \eta = \frac{\mu\sqrt{(m-1)^2+4mC}+\mu(m-1)}{2},\ \ \textit{and}\ \ \alpha = 1-\frac{1}{(m+1)+\sqrt{(m-1)^2+4mC}}, \tag{2.3.26}$$

where

$$C = \frac{8L}{\mu}. \tag{2.3.27}$$

Then for any $k \geq 1$, *we have*

$$\mathbb{E}[P(x^k, x^*)] \leq (1 + \tfrac{3L_f}{\mu})\alpha^k P(x^0, x^*), \tag{2.3.28}$$

$$\mathbb{E}[\Psi(\bar{x}^k) - \Psi^*] \leq \alpha^{k/2}(1-\alpha)^{-1}\left[\mu + 2L_f + \tfrac{L_f^2}{\mu}\right] P(x^0, x^*). \tag{2.3.29}$$

As a consequence, the number of iterations performed by the RPDG method to find a stochastic ϵ*-solution and a stochastic* (ϵ, λ)*-solution of (1.1.1), in terms of the distance to the optimal solution, i.e.,* $\mathbb{E}[P(x^k, x^*)]$, *can be bounded by* $K(\epsilon, C)$ *and* $K(\lambda\epsilon, C)$, *respectively, where*

$$K(\epsilon, C) := \left[(m+1) + \sqrt{(m-1)^2 + 4mC}\right] \log\left[(1 + \tfrac{3L_f}{\mu})\tfrac{P(x^0, x^*)}{\epsilon}\right]. \tag{2.3.30}$$

Similarly, the total number of iterations performed by the RPDG method to find a stochastic ϵ*-solution and a stochastic* (ϵ, λ)*-solution of (1.1.1), in terms of the primal optimality gap, i.e.,* $\mathbb{E}[\Psi(\bar{x}^k) - \Psi^*]$, *can be bounded by* $\tilde{K}(\epsilon, C)$ *and* $\tilde{K}(\lambda\epsilon, C)$, *respectively, where*

$$\tilde{K}(\epsilon, C) := 2\left[(m+1) + \sqrt{(m-1)^2 + 4mC}\right] \log\left[2(\mu + 2L_f + \tfrac{L_f^2}{\mu})(m + \sqrt{mC})\tfrac{P(x^0, x^*)}{\epsilon}\right]. \tag{2.3.31}$$

Proof. It follows from (2.3.26) that

$$(1-\alpha)(1+\tau) = 1/(2m) \leq p_i,\ \ (1-\alpha)\eta = (\alpha - 1/2)\mu \leq \alpha\mu,\ \ \text{and}\ \ \eta\tau p_i = \mu C p_i \geq 4L_i/m,$$

and hence that the conditions in (2.3.20)-(2.3.22) are satisfied. Notice that by the fact that $\alpha \geq 3/4, \ \ \forall m \geq 1$ and (2.3.26), we have

$$1 + \tfrac{L_f \alpha}{(1-\alpha)\eta} = 1 + L_f \tfrac{\alpha}{(\alpha - 1/2)\mu} \leq 1 + \tfrac{3L_f}{\mu}.$$

Using the above bound in (2.3.23), we obtain (2.3.28). It follows from the facts $(1-\alpha)\eta \leq \alpha\mu$, $1/2 \leq \alpha \leq 1, \forall m \geq 1$, and $\eta \geq \mu\sqrt{C} > 2\mu$ that

$$\alpha^{-1}\eta + \tfrac{3-2\alpha}{1-\alpha} L_f + \tfrac{2L_f^2 \alpha}{(1-\alpha)\eta} \leq (1-\alpha)^{-1}(\mu + 2L_f + \tfrac{L_f^2}{\mu}).$$

Using the above bound in (2.3.24), we obtain (2.3.29). Denoting $D \equiv (1 + \frac{3L_f}{\mu}) P(x^0, x^*)$, we conclude from (2.3.28) and the fact that $\log x \leq x - 1$ for any $x \in (0, 1)$ that

$$\mathbb{E}[P(x^{K(\epsilon,C)}, x^*)] \leq D\alpha^{\frac{\log(D/\epsilon)}{1-\alpha}} \leq D\alpha^{\frac{\log(D/\epsilon)}{-\log \alpha}} = D\alpha^{\frac{\log(\epsilon/D)}{\log \alpha}} = \epsilon.$$

Moreover, by Markov's inequality, (2.3.28) and the fact that $\log x \leq x - 1$ for any $x \in (0, 1)$, we have

$$\text{Prob}\{P(x^{K(\lambda\epsilon,C)}, x^*) > \epsilon\} \leq \tfrac{1}{\epsilon}\mathbb{E}[P(x^{K(\lambda\epsilon,C)}, x^*)] \leq \tfrac{D}{\epsilon}\alpha^{\frac{\log(D/(\lambda\epsilon))}{1-\alpha}} \leq \tfrac{D}{\epsilon}\alpha^{\frac{\log(\lambda\epsilon/D)}{\log \alpha}} = \lambda.$$

The proofs for the complexity bounds in terms of the primal optimality gap is similar and hence the details are skipped. ■

The non-uniform distribution in (2.3.25) requires the estimation of the Lipschitz constants L_i, $i = 1, \ldots, m$. In case such information is not available, we can use a uniform distribution for i_t, and as a result, the complexity bounds will depend on a larger condition number given by $\max_{i=1,\ldots,m} L_i/\mu$. However, if we do have $L_1 = L_2 = \cdots = L_m$, then the results obtained by using a uniform distribution is slightly sharper than the one by using a non-uniform distribution in Corollary 2.3.5.

Corollary 2.3.6 *Suppose that* $\{i_t\}$ *in the RPDG method are uniformly distributed over* $\{1, \ldots, m\}$ *according to*

$$p_i = \text{Prob}\{i_t = i\} = \tfrac{1}{m}, i = 1, \ldots, m. \tag{2.3.32}$$

Also assume that $\{\tau_t\}$, $\{\eta_t\}$, *and* $\{\alpha_t\}$ *are set to (2.3.19) with*

$$\tau = \tfrac{\sqrt{(m-1)^2+4m\bar{C}}-(m-1)}{2m}, \ \eta = \tfrac{\mu\sqrt{(m-1)^2+4m\bar{C}}+\mu(m-1)}{2}, \ \textit{and} \ \alpha = 1-\tfrac{2}{(m+1)+\sqrt{(m-1)^2+4m\bar{C}}}, \tag{2.3.33}$$

where

$$\bar{C} := \tfrac{4}{\mu} \max_{i=1,\ldots,m} L_i. \tag{2.3.34}$$

Then we have

$$\mathbb{E}[P(x^k, x^*)] \leq (1 + \tfrac{L_f}{\mu})\alpha^k P(x^0, x^*), \tag{2.3.35}$$

$$\mathbb{E}[\Psi(\bar{x}^k) - \Psi^*] \leq \alpha^{k/2}(1-\alpha)^{-1}\left(\mu + 2L_f + \tfrac{L_f^2}{\mu}\right) P(x^0, x^*). \tag{2.3.36}$$

for any $k \geq 1$. *As a consequence, the number of iterations performed by the RPDG method to find a stochastic* ϵ*-solution and a stochastic* (ϵ, λ)*-solution of (1.1.1), in terms of the distance to the optimal solution, i.e.,* $\mathbb{E}[P(x^k, x^*)]$, *can be bounded by* $K_u(\epsilon, \bar{C})$ *and* $K_u(\lambda\epsilon, \bar{C})$, *respectively, where*

$$K_u(\epsilon, \bar{C}) := \tfrac{(m+1)+\sqrt{(m-1)^2+4m\bar{C}}}{2} \log\left[(1 + \tfrac{L_f}{\mu})\tfrac{P(x^0,x^*)}{\epsilon}\right].$$

Similarly, the total number of iterations performed by the RPDG method to find a stochastic ϵ*-solution and a stochastic* (ϵ, λ)*-solution of (1.1.1), in terms of the primal optimality gap, i.e.,* $\mathbb{E}[\Psi(\bar{x}^k) - \Psi^*]$, *can be bounded by* $\tilde{K}(\epsilon, \bar{C})/2$ *and* $\tilde{K}(\lambda\epsilon, \bar{C})/2$, *respectively, where* $\tilde{K}(\epsilon, \bar{C})$ *is defined in (2.3.31).*

Proof. It follows from (2.3.33) that

$$(1-\alpha)(1+\tau) = 1/m = p_i, \;\; (1-\alpha)\eta - \alpha\mu = 0, \;\; \text{and} \;\; \eta\tau = \mu\bar{C} \geq 4L_i,$$

and hence that the conditions in (2.3.20)-(2.3.22) are satisfied. By the identity $(1-\alpha)\eta = \alpha\mu$, we have

$$1 + \tfrac{L_f\alpha}{(1-\alpha)\eta} = 1 + \tfrac{L_f}{\mu}.$$

Using the above bound in (2.3.23), we obtain (2.3.35). Moreover, note that $\eta \geq \mu\sqrt{\bar{C}} \geq 2\mu$ and $2/3 \leq \alpha \leq 1, \forall m \geq 1$ we have

$$\alpha^{-1}\eta + \tfrac{3-2\alpha}{1-\alpha}L_f + \tfrac{2L_f^2\alpha}{(1-\alpha)\eta} \leq (1-\alpha)^{-1}(\mu + 2L_f + \tfrac{L_f^2}{\mu}).$$

Using the above bound in (2.3.24), we obtain (2.3.36). The proofs for the complexity bounds are similar to those in Corollary 2.3.5 and hence the details are skipped. ∎

Comparing the complexity bounds obtained from Corollaries 2.3.5 and 2.3.6 with those of any optimal deterministic first-order method, they differ in a factor of $\mathcal{O}(\sqrt{mL_f/L})$, whenever $\sqrt{mC}\log(1/\epsilon)$ is dominating in (2.3.30). Clearly, when L_f and L are in the same order of magnitude, RPDG can save up to $\mathcal{O}(\sqrt{m})$ gradient evaluations for the component function f_i than the deterministic first-order methods. However, it should be pointed out that L_f can be much smaller than L. In particular, in some cases we may have $L_i = L_j, \forall i, j \in \{1, \ldots, m\}$, and $L_f = L/m$. In the next subsection, we will construct examples in such extreme cases to obtain the lower complexity bound for general randomized incremental gradient methods.

2.3.3 Lower Complexity Bound for Randomized Methods

Our goal in this subsection is to demonstrate that the complexity bounds obtained in Theorem 2.3.4, and Corollaries 2.3.5 and 2.3.6 for the RPDG method are essentially not improv-

able. Observe that although there exist rich lower complexity bounds in the literature for deterministic first-order methods (e.g. [23, 6]), the study on lower complexity bounds for randomized methods are still quite limited. Recently Agarwal and Bottou [32] suggested a lower complexity bound for minimizing the finite-sum convex optimization problem given in the form of (1.1.1). However, their bounds are developed for deterministic algorithms and hence not applicable to randomized incremental gradient methods.

To derive the performance limit of the incremental gradient methods described above, we consider a special class of unconstrained and separable strongly convex optimization problems given in the form of

$$\min_{x_i\in\mathbb{R}^{\tilde{n}},i=1,\ldots,m} \left\{\Psi(x) := \sum_{i=1}^m \left[\frac{1}{m}f_i(x_i) + \frac{\mu}{2}\|x_i\|_2^2\right]\right\}. \tag{2.3.37}$$

Here $\tilde{n} \equiv n/m \in \{1, 2, \ldots\}$ and $\|\cdot\|_2$ denotes standard Euclidean norm. To fix the notation, we also denote $x = (x_1, \ldots, x_m)$. Moreover, we assume that f_i's are quadratic functions given by

$$f_i(x_i) = \tfrac{\mu m(\mathcal{Q}-1)}{4}\left[\tfrac{1}{2}\langle Ax_i, x_i\rangle - \langle e_1, x_i\rangle\right], \tag{2.3.38}$$

where $e_1 := (1, 0, \ldots, 0)$ and A is a symmetric matrix in $\mathbb{R}^{\tilde{n}\times\tilde{n}}$ given by

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & \kappa \end{pmatrix} \quad \text{with} \quad \kappa = \tfrac{\sqrt{\mathcal{Q}}+3}{\sqrt{\mathcal{Q}}+1}. \tag{2.3.39}$$

Compared with the classic worst-case example given in [6], the tridiagonal matrix A above consists of a different diagonal element κ (instead of 2). This modification allows us to study problems of finite dimension more conveniently. It can be easily checked that $A \succeq 0$

and its maximum eigenvalue does not exceed 4. Indeed, for any $s \equiv (s_1, \ldots, s_{\tilde{n}}) \in \mathbb{R}^{\tilde{n}}$, we have

$$\begin{aligned}
\langle As, s\rangle &= s_1^2 + \textstyle\sum_{i=1}^{\tilde{n}-1}(s_i - s_{i+1})^2 + (\kappa - 1)s_{\tilde{n}}^2 \geq 0 \\
\langle As, s\rangle &\leq s_1^2 + \textstyle\sum_{i=1}^{\tilde{n}-1} 2(s_i^2 + s_{i+1}^2) + (\kappa - 1)s_{\tilde{n}}^2 \\
&= 3s_1^2 + 4\textstyle\sum_{i=2}^{\tilde{n}-1} s_i^2 + (\kappa + 1)s_{\tilde{n}}^2 \leq 4\|s\|_2^2,
\end{aligned}$$

where the last inequality follows from the fact that $\kappa \leq 3$. Therefore, for any $\mathcal{Q} > 1$, the component functions f_i in (2.3.38) are convex and their gradients are Lipschitz continuous with constant bounded by $L_i = m\mu(\mathcal{Q} - 1)$, $i = 1, \ldots, m$.

We consider a general class of randomized incremental gradient methods which sequentially acquire the gradient of a randomly selected component function f_{i_t} at iteration t. More specifically, we assume that the independent random variables i_t, $t = 1, 2, \ldots$, satisfy

$$\mathrm{Prob}\{i_t = i\} = p_i \quad \text{and} \quad \textstyle\sum_{i=1}^m p_i = 1, \;\; p_i \geq 0, i = 1, \ldots, m. \tag{2.3.40}$$

Similar to [6], we assume that these methods generate a sequence of test points $\{x^k\}$ such that

$$x^k \in x^0 + \mathrm{Lin}\{\nabla f_{i_1}(x^0), \ldots, \nabla f_{i_k}(x^{k-1})\}, \tag{2.3.41}$$

where Lin denotes the linear span.

Theorem 2.3.7 below describes the performance limit of the above randomized incremental gradient methods for solving (2.3.37).

Theorem 2.3.7 *Let x^* be the optimal solution of problem (2.3.37) and denote*

$$q := \tfrac{\sqrt{\mathcal{Q}}-1}{\sqrt{\mathcal{Q}}+1}. \tag{2.3.42}$$

Then the iterates $\{x^k\}$ generated by any randomized incremental gradient method must

satisfy

$$\frac{\mathbb{E}[\|x^k - x^*\|_2^2]}{\|x^0 - x^*\|_2^2} \geq \frac{1}{2}\exp\left(-\frac{4k\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2 - 4\sqrt{\mathcal{Q}}}\right) \tag{2.3.43}$$

for any $m \geq 2$ *and*

$$n \geq \underline{n}(m,k) \equiv \left(k + \frac{m}{2}\right)\frac{1}{\log(1/q)}. \tag{2.3.44}$$

As an immediate consequence of Theorem 2.3.7, we obtain a lower complexity bound for randomized incremental gradient methods.

Corollary 2.3.8 *The number of gradient evaluations performed by any randomized incremental gradient methods for finding a solution* $\bar{x} \in X$ *of problem (1.1.1) such that* $\mathbb{E}[\|\bar{x} - x^*\|_2^2] \leq \epsilon$ *cannot be smaller than*

$$\Omega\left\{\left(\sqrt{m\mathcal{C}} + m\right)\log\frac{\|x^0 - x^*\|_2^2}{\epsilon}\right\}$$

if n *is sufficiently large, where* $\mathcal{C} = L/\mu$ *and* $L = \frac{1}{m}\sum_{i=1}^m L_i$.

Proof. It follows from (2.3.43) that the number of iterations k required by any randomized incremental gradient methods to find an approximate solution $\bar{x}$ must satisfy

$$k \geq \left(\frac{m(\sqrt{\mathcal{Q}}+1)^2}{4\sqrt{\mathcal{Q}}} - 1\right)\log\frac{\|x^0 - x^*\|_2^2}{2\epsilon} \geq \left[\frac{m}{2}\left(\frac{\sqrt{\mathcal{Q}}}{2} + 1\right) - 1\right]\log\frac{\|x^0 - x^*\|_2^2}{2\epsilon}. \tag{2.3.45}$$

Noting that for the worst-case instance in (2.3.37), we have $L_i = m\mu(\mathcal{Q}-1)$, $i = 1, \ldots, m$, and hence that $L = \frac{1}{m}\sum_{i=1}^m L_i = m\mu(Q-1)$. Using this relation, we conclude that

$$k \geq \left[\frac{1}{2}\left(\frac{\sqrt{m\mathcal{C}+m^2}}{2} + m\right) - 1\right]\log\frac{\|x^0 - x^*\|_2^2}{2\epsilon} =: \underline{k}.$$

The above bound holds when $n \geq \underline{n}(m, \underline{k})$. ∎

In view of Theorem 2.3.7, we can also derive a lower complexity bound for randomized block coordinate descent methods, which update one randomly selected block of variables

at each iteration for $\min_{x\in X}\Psi(x)$. Here Ψ is smooth and strongly convex such that

$$\tfrac{\mu_\Psi}{2}\|x-y\|_2^2 \le \Psi(x)-\Psi(y)-\langle\nabla\Psi(y),x-y\rangle \le \tfrac{L_\Psi}{2}\|x-y\|_2^2, \forall x,y\in X.$$

Corollary 2.3.9 *The number of iterations performed by any randomized block coordinate descent methods for finding a solution $\bar{x}\in X$ of $\min_{x\in X}\Psi(x)$ such that $\mathbb{E}[\|\bar{x}-x^*\|_2^2]\le\epsilon$ cannot be smaller than*

$$\Omega\left\{\left(m\sqrt{\mathcal{Q}_\Psi}\right)\log\tfrac{\|x^0-x^*\|_2^2}{\epsilon}\right\}$$

if n is sufficiently large, where $\mathcal{Q}_\Psi = L_\Psi/\mu_\Psi$ denotes the condition number of Ψ.

Proof. The worst-case instances in (2.3.37) have a block separable structure. Therefore, any randomized incremental gradient methods are equivalent to randomized block coordinate descent methods. The result then immediately follows from (2.3.45). ∎

2.4 Generalization of Randomized Primal-dual Gradient Methods

In this section, we generalize the RPDG method for solving a few different types of convex optimization problems which are not necessarily smooth and strongly convex.

2.4.1 Smooth Problems with Bounded Feasible Sets

Our goal in this subsection is to generalize RPDG for solving smooth problems without strong convexity (i.e., $\mu = 0$). Different from the deterministic PDG method, it is difficult to develop a simple stepsize policy for $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ which can guarantee the convergence of this method unless a weaker termination criterion is used (see [26]). In order to obtain stronger convergence results, we will discuss a different approach obtained by applying the RPDG method to a slightly perturbed problem of (1.1.1).

In order to apply this perturbation approach, we will assume that X is bounded (see

Subsection 2.4.3 for possible extensions), i.e., given $x_0 \in X$, $\exists \Omega_X \geq 0$ s.t.

$$\max_{x \in X} P_\omega(x_0, x) \leq \Omega_X^2. \tag{2.4.1}$$

Now we define the perturbation problem as

$$\Psi_\delta^* := \min_{x \in X} \{\Psi_\delta(x) := f(x) + h(x) + \delta P_\omega(x_0, x)\}, \tag{2.4.2}$$

for some fixed $\delta > 0$. It is well-known that an approximate solution of (2.4.2) will also be an approximate solution of (1.1.1) if δ is sufficiently small. More specifically, it is easy to verify that

$$\Psi^* \leq \Psi_\delta^* \leq \Psi^* + \delta \Omega_X^2, \tag{2.4.3}$$

$$\Psi(x) \leq \Psi_\delta(x) \leq \Psi(x) + \delta \Omega_X^2, \ \ \forall x \in X. \tag{2.4.4}$$

The following result describes the complexity associated with this perturbation approach for solving smooth problems without strong convexity (i.e., $\mu = 0$).

Proposition 2.4.10 *Let us apply the RPDG method with the parameter settings in Corollary 2.3.5 to the perturbation problem (2.4.2) with*

$$\delta = \tfrac{\epsilon}{2\Omega_X^2}, \tag{2.4.5}$$

for some $\epsilon > 0$. Then we can find a solution $\bar{x} \in X$ s.t. $\mathbb{E}[\Psi(\bar{x}) - \Psi^*] \leq \epsilon$ in at most

$$\mathcal{O}\left\{\left(m + \sqrt{\tfrac{mL\Omega_X^2}{\epsilon}}\right) \log \tfrac{mL_f \Omega_X}{\epsilon}\right\} \tag{2.4.6}$$

iterations. Moreover, we can find a solution $\bar{x} \in X$ s.t. Prob$\{\Psi(\bar{x}) - \Psi^* > \epsilon\} \leq \lambda$ *for any*

$\lambda \in (0,1)$ *in at most*

$$\mathcal{O}\left\{\left(m+\sqrt{\tfrac{mL\Omega_X^2}{\epsilon}}\right)\log\tfrac{mL_f\Omega_X}{\lambda\epsilon}\right\} \tag{2.4.7}$$

iterations.

Proof. Let x_δ^* be the optimal solution of (2.4.2). Denote $C := 16L\Omega_X^2/\epsilon$ and

$$K := 2\left[(m+1)+\sqrt{(m-1)^2+4mC}\right]\log\left[(m+\sqrt{mC})(\delta+2L_f+\tfrac{L_f^2}{\delta})\tfrac{4\Omega_X^2}{\epsilon}\right].$$

It can be easily seen that

$$\Psi(\bar{x}^K)-\Psi^* \le \Psi_\delta(\bar{x}^K)-\Psi_\delta^*+\delta\Omega_X^2 = \Psi_\delta(\bar{x}^K)-\Psi_\delta^*+\tfrac{\epsilon}{2}.$$

Note that problem (2.4.2) is given in the form of (1.1.1) with the strongly convex modulus $\mu=\delta$, and $h(x)=h(x)-\delta\langle\omega'(x_0),x\rangle$. Hence by applying Corollary 2.3.5, we have

$$\mathbb{E}[\Psi_\delta(\bar{x}^K)-\Psi_\delta^*] \le \tfrac{\epsilon}{2}.$$

Combining these two inequalities, we have $\mathbb{E}[\Psi(\bar{x}^K)-\Psi^*]\le\epsilon$, which implies the bound in (2.4.6). The bound in (2.4.7) can be shown similarly and hence the details are skipped. ■

Note that in [96], Zhu and Hazan proposed a method with a diminishing perturbation term to obtain the same rate of convergence, $\mathcal{O}(\sqrt{m/\epsilon})$ (regardless of some logarithmic factors), which can also be applied to the RPDG method so that we do not need to fix ϵ before running the algorithm.

Observe that if we apply a deterministic optimal first-order method (e.g., Nesterov's method or the PDG method), the total number of gradient evaluations for $\nabla f_i, i=1,\ldots,m$, would be given by

$$m\sqrt{\tfrac{L_f\Omega_X^2}{\epsilon}}.$$

Comparing this bound with (2.4.6), we can see that the number of gradient evaluations performed by the RPDG method can be $\mathcal{O}\left(\sqrt{m}\log^{-1}(mL_f\Omega_X/\epsilon)\right)$ times smaller than these deterministic methods when L and L_f are in the same order of magnitude.

2.4.2 Structured Nonsmooth Problems

In this subsection, we assume that the smooth components f_i are nonsmooth but can be approximated closely by smooth ones. More specifically, we assume that

$$f_i(x) := \max_{y_i \in Y_i} \langle A_i x, y_i \rangle - q_i(y_i). \tag{2.4.8}$$

Nesterov in an important work [97] shows that we can approximate $f_i(x)$ and f, respectively, by

$$\tilde{f}_i(x,\delta) := \max_{y_i \in Y_i} \langle A_i x, y_i \rangle - q_i(y_i) - \delta v_i(y_i) \ \text{ and } \ \tilde{f}(x,\delta) = \tfrac{1}{m}\textstyle\sum_{i=1}^m \tilde{f}_i(x,\delta), \tag{2.4.9}$$

where $v_i(y_i)$ is a strongly convex function with modulus 1 such that

$$0 \le v_i(y_i) \le \Omega_{Y_i}^2, \quad \forall y_i \in Y_i. \tag{2.4.10}$$

In particular, we can easily show that

$$\tilde{f}_i(x,\delta) \le f_i(x) \le \tilde{f}_i(x,\delta) + \delta\Omega_{Y_i}^2 \ \text{ and } \ \tilde{f}(x,\delta) \le f(x) \le \tilde{f}(x,\delta) + \delta\Omega_Y^2, \tag{2.4.11}$$

for any $x \in X$, where $\Omega_Y^2 = \frac{1}{m}\sum_{i=1}^m \Omega_{Y_i}^2$. Moreover, $f_i(\cdot,\delta)$ and $f(\cdot,\delta)$ are continuously differentiable and their gradients are Lipschitz continuous with constants given by

$$\tilde{L}_i = \frac{\|A_i\|^2}{\delta} \quad \text{and} \quad \tilde{L} = \frac{\sum_{i=1}^m \|A_i\|^2}{m\delta} = \frac{\|A\|^2}{m\delta}, \tag{2.4.12}$$

respectively. As a consequence, we can apply the RPDG method to solve the approximation problem

$$\tilde{\Psi}_\delta^* := \min_{x\in X}\left\{\tilde{\Psi}_\delta(x) := \tilde{f}(x,\delta) + h(x) + \mu\omega(x)\right\}. \tag{2.4.13}$$

The following result provides complexity bounds of the RPDG method for solving the above structured nonsmooth problems for the case when $\mu > 0$.

Proposition 2.4.11 *Let us apply the RPDG method with the parameter settings in Corollary 2.3.5 to the approximation problem (2.4.13) with*

$$\delta = \tfrac{\epsilon}{2\Omega_Y^2}, \tag{2.4.14}$$

for some $\epsilon > 0$. Then we can find a solution $\bar{x} \in X$ s.t. $\mathbb{E}[\Psi(\bar{x}) - \Psi^*] \le \epsilon$ in at most

$$\mathcal{O}\left\{\|A\|\Omega_Y\sqrt{\tfrac{m}{\mu\epsilon}}\log\tfrac{\|A\|\Omega_X\Omega_Y}{m\mu\epsilon}\right\} \tag{2.4.15}$$

iterations. Moreover, we can find a solution $\bar{x} \in X$ s.t. Prob$\{\Psi(\bar{x}) - \Psi^* > \epsilon\} \le \lambda$ *for any $\lambda \in (0,1)$ in at most*

$$\mathcal{O}\left\{\|A\|\Omega_Y\sqrt{\tfrac{m}{\mu\epsilon}}\log\tfrac{\|A\|\Omega_X\Omega_Y}{\lambda m\mu\epsilon}\right\} \tag{2.4.16}$$

iterations.

Proof. It follows from (2.4.11) and (2.4.13) that

$$\Psi(\bar{x}^k) - \Psi^* \le \tilde{\Psi}_\delta(\bar{x}^k) - \tilde{\Psi}_\delta^* + \delta\Omega_Y^2 = \tilde{\Psi}_\delta(\bar{x}^k) - \tilde{\Psi}_\delta^* + \tfrac{\epsilon}{2}. \tag{2.4.17}$$

Using relation (2.4.12) and Corollaries 2.3.5, we conclude that a solution $\bar{x}^k \in X$ satisfying $\mathbb{E}[\tilde{\Psi}_\delta(\bar{x}^k) - \tilde{\Psi}_\delta^*] \le \epsilon/2$ can be found in

$$\mathcal{O}\left\{\|A\|\Omega_Y\sqrt{\tfrac{m}{\mu\epsilon}}\log\left[(m+\sqrt{\tfrac{m\tilde{L}}{\mu}})\left(\mu + 2\tilde{L} + \tfrac{\tilde{L}^2}{\mu}\right)\tfrac{\Omega_X^2}{\epsilon}\right]\right\}$$

iterations. This observation together with (2.4.17) and the definition of $\tilde{L}$ in (2.4.12) then imply the bound in (2.4.15). The bound in (2.4.16) follows similarly from (2.4.17) and Corollaries 2.3.5, and hence the details are skipped. ■

The following result holds for the RPDG method applied to the above structured nonsmooth problems when $\mu = 0$.

Proposition 2.4.12 *Let us apply the RPDG method with the parameter settings in Corollary 2.3.5 to the approximation problem (2.4.13) with δ in (2.4.14) for some $\epsilon > 0$. Then we can find a solution $\bar{x} \in X$ s.t. $\mathbb{E}[\Psi(\bar{x}) - \Psi^*] \leq \epsilon$ in at most*

$$\mathcal{O}\left\{\frac{\sqrt{m}\|A\|_{\Omega_X\Omega_Y}}{\epsilon}\log\frac{\|A\|_{\Omega_X\Omega_Y}}{m\epsilon}\right\}$$

iterations. Moreover, we can find a solution $\bar{x} \in X$ s.t. Prob$\{\Psi(\bar{x}) - \Psi^* > \epsilon\} \leq \lambda$ *for any $\lambda \in (0, 1)$ in at most*

$$\mathcal{O}\left\{\frac{\sqrt{m}\|A\|_{\Omega_X\Omega_Y}}{\epsilon}\log\frac{m\|A\|_{\Omega_X\Omega_Y}}{\lambda m\epsilon}\right\}$$

iterations.

Proof. Similarly to the arguments used in the proof of Proposition 2.4.11, our results follow from (2.4.17), and an application of Proposition 2.4.10 to problem (2.4.13). ■

By Propositions 2.4.11 and 2.4.12, the total number of gradient computations for $\tilde{f}(\cdot, \delta)$ performed by the RPDG method, after disregarding the logarithmic factors, can be $\mathcal{O}(\sqrt{m})$ times smaller than those required by deterministic first-order methods, such as Nesterov's smoothing technique [97].

2.4.3 Unconstrained Smooth Problems

In this subsection, we set $X = \mathbb{R}^n$, $h(x) = 0$, and $\mu = 0$ in (1.1.1) and consider the basic convex programming problem of

$$f^* := \min_{x \in \mathbb{R}^n} \left\{ f(x) := \frac{1}{m} \sum_{i=1}^m f_i(x) \right\}. \tag{2.4.18}$$

We assume that the set of optimal solutions X^* of this problem is nonempty.

We will still use the perturbation-based approach as described in Subsection 2.4.1 by solving the perturbation problem given by

$$f_\delta^* := \min_{x \in \mathbb{R}^n} \left\{ f_\delta(x) := f(x) + \frac{\delta}{2} \|x - x^0\|_2^2, \right\}, \tag{2.4.19}$$

for some $x^0 \in X, \delta > 0$, where $\|\cdot\|_2$ denotes the Euclidean norm. Also let L_δ denote the Lipschitz constant for $f_\delta(x)$. Clearly, $L_\delta = L + \delta$. Since the problem is unconstrained and the information on the size of the optimal solution is unavailable, it is hard to estimate the total number of iterations by using the absolute accuracy in terms of $\mathbb{E}[f(\bar{x}) - f^*]$. Instead, we define the relative accuracy [2] associated with a given $\bar{x} \in X$ by

$$R_{ac}(\bar{x}, x^0, f^*) := \frac{2[f(\bar{x}) - f^*]}{L(1 + \min_{u \in X^*} \|x^0 - u\|_2^2)}. \tag{2.4.20}$$

We are now ready to establish the complexity of the RPDG method applied to (2.4.18) in terms of $R_{ac}(\bar{x}, x^0, f^*)$.

Proposition 2.4.13 *Let us apply the RPDG method with the parameter settings in Corollary 2.3.5 to the perturbation problem (2.4.19) with*

$$\delta = \frac{L\epsilon}{2}, \tag{2.4.21}$$

[2] Relative accuracy is a common termination criteria for unconstrained problems, see [98] for a similar example.

for some $\epsilon > 0$. Then we can find a solution $\bar{x} \in X$ s.t. $\mathbb{E}[R_{ac}(\bar{x}, x^0, f^*)] \leq \epsilon$ in at most

$$\mathcal{O}\left\{\sqrt{\tfrac{m}{\epsilon}} \log \tfrac{m}{\epsilon}\right\} \tag{2.4.22}$$

iterations. Moreover, we can find a solution $\bar{x} \in X$ s.t. Prob$\{R_{ac}(\bar{x}, x^0, f^*) > \epsilon\} \leq \lambda$ *for any $\lambda \in (0, 1)$ in at most*

$$\mathcal{O}\left\{\sqrt{\tfrac{m}{\epsilon}} \log \tfrac{m}{\lambda\epsilon}\right\} \tag{2.4.23}$$

iterations.

Proof. Let x_δ^* be the optimal solution of (2.4.19). Also let x^* be the optimal solution of (2.4.18) that is closest to x^0, i.e., $x^* = \operatorname{argmin}_{u \in X^*} \|x^0 - u\|_2$. It then follows from the strong convexity of f_δ that

$$\begin{aligned}
\tfrac{\delta}{2}\|x_\delta^* - x^*\|_2^2 &\leq f_\delta(x^*) - f_\delta(x_\delta^*) \\
&= f(x^*) + \tfrac{\delta}{2}\|x^* - x^0\|_2^2 - f_\delta(x_\delta^*) \\
&\leq \tfrac{\delta}{2}\|x^* - x^0\|_2^2,
\end{aligned}$$

which implies that

$$\|x_\delta^* - x^*\|_2 \leq \|x^* - x^0\|_2. \tag{2.4.24}$$

Moreover, using the definition of f_δ and the fact that x^* is feasible to (2.4.19), we have

$$f^* \leq f_\delta^* \leq f^* + \tfrac{\delta}{2}\|x^* - x^0\|_2^2,$$

which implies that

$$\begin{aligned}
f(\bar{x}^K) - f^* &\leq f_\delta(\bar{x}^K) - f_\delta^* + f_\delta^* - f^* \\
&\leq f_\delta(\bar{x}^K) - f_\delta^* + \tfrac{\delta}{2}\|x^* - x^0\|_2^2.
\end{aligned}$$

Now suppose that we run the RPDG method applied to (2.4.19) for K iterations. Then by Corollary 2.3.5, we have

$$\begin{aligned}\mathbb{E}[f_\delta(\bar{x}^K) - f_\delta^*] &\le \alpha^{K/2}(1-\alpha)^{-1}\left(\delta + 2L_\delta + \tfrac{L_\delta^2}{\delta}\right)\|x^0 - x_\delta^*\|_2^2 \\ &\le \alpha^{K/2}(1-\alpha)^{-1}\left(\delta + 2L_\delta + \tfrac{L_\delta^2}{\delta}\right)[\|x^0 - x^*\|_2^2 + \|x^* - x_\delta^*\|_2^2] \\ &= 2\alpha^{K/2}(1-\alpha)^{-1}\left(3\delta + 2L + \tfrac{(L+\delta)^2}{\delta}\right)\|x^0 - x^*\|_2^2,\end{aligned}$$

where the last inequality follows from (2.4.24) and α is defined in (2.3.26) with $C = 8L_\delta/\delta = \frac{8(L+\delta)}{\delta} = 8(2/\epsilon + 1)$. Combining the above two relations, we have

$$\mathbb{E}[f(\bar{x}^K) - f^*] \le \left[2\alpha^{K/2}(1-\alpha)^{-1}\left(3\delta + 2L + \tfrac{(L+\delta)^2}{\delta}\right) + \tfrac{\delta}{2}\right][\|x^0 - x^*\|_2^2.$$

Dividing both sides of the above inequality by $L(1 + \|x^0 - x^*\|_2^2)/2$, we obtain

$$\begin{aligned}\mathbb{E}[R_{ac}(\bar{x}^K, x^0, f^*)] &\le \tfrac{2}{L}\left[2\alpha^{K/2}(1-\alpha)^{-1}\left(3\delta + 2L + \tfrac{(L+\delta)^2}{\delta}\right) + \tfrac{\delta}{2}\right] \\ &\le 4\left(m + 2\sqrt{2m(\tfrac{2}{\epsilon}+1)}\right)\left(3\epsilon + 4 + (2+\epsilon)(\tfrac{2}{\epsilon}+1)\right)\alpha^{K/2} + \tfrac{\epsilon}{2},\end{aligned}$$

which clearly implies the bound in (2.4.22). The bound in (2.4.23) also follows from the above inequality and the Markov's inequality. ∎

By Proposition 2.4.13, the total number of gradient evaluations for the component functions f_i required by the RPDG method can be $\mathcal{O}(\sqrt{m}\log^{-1}(m/\epsilon))$ times smaller than those performed by deterministic optimal first-order methods.

2.5 Complexity Analysis

Our main goal in this section is to prove the main theorems in Sections 2.2 and 2.3. After introducing some basic tools and general results about PDG and RPDG methods in Subsection 2.5.1 and 2.5.2, respectively, we provide the proofs for Theorem 2.2.3 and

Theorem 2.3.4, which describe the main convergence properties for the PDG and RPDG methods, in Subsection 2.5.3. Moreover, in Subsection 2.5.4, we provide the proof for the lower complexity bound in Theorem 2.3.7.

2.5.1 Some Basic Tools

The following result provides a few different bounds on the diameter of the dual feasible sets $\mathcal{G}$ and $\mathcal{Y}$ in (2.2.7) and (2.3.1).

Lemma 2.5.14 *Let $x^0 \in X$ be given, $y_i^0 = \frac{1}{m}\nabla f_i(x^0)$, $i = 1, \ldots, m$, and $g^0 = \nabla f(x^0)$. Assume that $J_i'(y_i^0) = x^0$ and $J_f'(g^0) = x^0$ in the definition of $D(y^0, y)$ and $D_f(g^0, g)$ in (2.3.4) and (2.2.5), respectively.*

a) For any $x \in X$ and $y_i = \frac{1}{m}\nabla f_i(x)$, $i = 1, \ldots, m$, we have

$$D(y^0, y) \leq \frac{L_f}{2}\|x^0 - x\|^2 \leq L_f P(x^0, x). \tag{2.5.1}$$

b) If $x^* \in X$ is an optimal solution of (1.1.1) and $y_i^* = \frac{1}{m}\nabla f_i(x^*)$, $i = 1, \ldots, m$, then

$$D(y^0, y^*) \leq \Psi(x^0) - \Psi(x^*). \tag{2.5.2}$$

c) For any $x \in X$ and $g = \nabla f(x)$, we have

$$D_f(g^0, g) \leq \frac{L_f}{2}\|x^0 - x\|^2. \tag{2.5.3}$$

Proof. We first show part a). It follows from the definitions of $D(y^0, y)$ and J_i, that

$$\begin{aligned} D(y^0, y) &= J(y) - J(y^0) - \textstyle\sum_{i=1}^m \langle J_i'(y_i^0), y_i - y_i^0\rangle \\ &= \langle x, Uy\rangle - f(x) + f(x^0) - \langle x^0, Uy^0\rangle - \langle x^0, U(y - y^0)\rangle \\ &= f(x^0) - f(x) - \langle Uy, x^0 - x\rangle \end{aligned}$$

$$\leq \frac{L_f}{2}\|x^0 - x\|^2 \leq L_f P(x^0, x),$$

where the last inequality follows from (2.2.2). We now show part b). By the above relation, the convexity of h and ω, and the optimality of (x^*, y^*), we have

$$\begin{aligned} D(y^0, y^*) &= f(x^0) - f(x^*) - \langle Uy^*, x^0 - x^*\rangle \\ &= f(x^0) - f(x^*) + \langle h'(x^*) + \mu\omega'(x^*), x^0 - x^*\rangle - \langle Uy^* + h'(x^*) + \mu\omega'(x^*), x^0 - x^*\rangle \\ &\leq f(x^0) - f(x^*) + \langle h'(x^*) + \mu\omega'(x^*), x^0 - x^*\rangle \leq \Psi(x^0) - \Psi(x^*). \end{aligned}$$

The proof of part c) is similar to part a) and hence the details are skipped. ∎

The following lemma gives an important bound for the primal optimality gap $\Psi(\bar{x}) - \Psi(x^*)$ for some $\bar{x} \in X$.

Lemma 2.5.15 *Let $(\bar{x}, \bar{y}) \in Z$ be a given pair of feasible solutions of (2.3.1), and $z^* = (x^*, y^*)$ be a pair of optimal solutions of (2.3.1). Then, we have*

$$\Psi(\bar{x}) - \Psi(x^*) \leq Q((\bar{x}, \bar{y}), z^*) + \frac{L_f}{2}\|\bar{x} - x^*\|^2. \tag{2.5.4}$$

Proof. Let $\bar{y}_* = (\frac{1}{m}\nabla f_1(\bar{x}); \frac{1}{m}\nabla f_2(\bar{x}); \ldots; \frac{1}{m}\nabla f_m(\bar{x}))$, and by the definition of $Q(\cdot, \cdot)$ in (2.3.3), we have

$$\begin{aligned} Q((\bar{x}, \bar{y}), z^*) &= [h(\bar{x}) + \mu\omega(\bar{x}) + \langle \bar{x}, Uy^*\rangle - J(y^*)] - [h(x^*) + \mu\omega(x^*) + \langle x^*, U\bar{y}\rangle - J(\bar{y})] \\ &\geq [h(\bar{x}) + \mu\omega(\bar{x}) + \langle \bar{x}, U\bar{y}_*\rangle - J(\bar{y}_*)] + \langle \bar{x}, U(y^* - \bar{y}_*)\rangle - J(y^*) + J(\bar{y}_*) \\ &\quad - \left[h(x^*) + \mu\omega(x^*) + \max_{y\in\mathcal{Y}} \{\langle x^*, Uy\rangle - J(y)\}\right] \\ &= \Psi(\bar{x}) - \Psi(x^*) + \langle \bar{x}, U(y^* - \bar{y}_*)\rangle - \langle x^*, Uy^*\rangle + f(x^*) + \langle \bar{x}, U\bar{y}_*\rangle - f(\bar{x}) \\ &= \Psi(\bar{x}) - \Psi(x^*) + f(x^*) - f(\bar{x}) + \langle \bar{x} - x^*, \nabla f(x^*)\rangle \geq \Psi(\bar{x}) - \Psi(x^*) - \frac{L_f}{2}\|\bar{x} - x^*\|^2, \end{aligned}$$

where the second equality follows from the fact that $J_i, i = 1, \ldots, m$, are the conjugate

functions of f_i. ■

2.5.2 General Results for Both PDG and RPDG

We will establish some general convergence results in Proposition 2.5.19 which holds for both deterministic and randomized PDG methods by viewing PDG as a special case of RPDG with $m = 1$. Then both Theorems 2.2.3 and 2.3.4 follow as some immediate consequences of Proposition 2.5.19.

Before showing Proposition 2.5.19 we will develop a few technical results. Lemma 2.5.16 below characterizes the solutions of the prox-mapping in (2.2.3) and (2.3.5). This result generalizes some previous results (e.g., Lemma 6 of [99] and Lemma 2 of [11]).

Lemma 2.5.16 *Let U be a closed convex set and a point $\tilde{u} \in U$ be given. Also let $w : U \to \mathbb{R}$ be a convex function and*

$$W(\tilde{u}, u) = w(u) - w(\tilde{u}) - \langle w'(\tilde{u}), u - \tilde{u} \rangle, \tag{2.5.5}$$

for some $w'(\tilde{u}) \in \partial w(\tilde{u})$. Assume that the function $q : U \to \mathbb{R}$ satisfies

$$q(u_1) - q(u_2) - \langle q'(u_2), u_1 - u_2 \rangle \geq \mu_0 W(u_2, u_1), \ \ \forall u_1, u_2 \in U \tag{2.5.6}$$

for some $\mu_0 \geq 0$. Also assume that the scalars μ_1 and μ_2 are chosen such that $\mu_0+\mu_1+\mu_2 \geq 0$. If

$$u^* \in \operatorname{Argmin}\{q(u) + \mu_1 w(u) + \mu_2 W(\tilde{u}, u) : u \in U\}, \tag{2.5.7}$$

then for any $u \in U$, we have

$$q(u^*) + \mu_1 w(u^*) + \mu_2 W(\tilde{u}, u^*) + (\mu_0 + \mu_1 + \mu_2) W(u^*, u) \leq q(u) + \mu_1 w(u) + \mu_2 W(\tilde{u}, u).$$

Proof. Let $\phi(u) := q(u) + \mu_1 w(u) + \mu_2 W(\tilde{u}, u)$. It can be easily checked that for any

$u_1, u_2 \in U$,

$$W(\tilde{u}, u_1) = W(\tilde{u}, u_2) + \langle W'(\tilde{u}, u_2), u_1 - u_2 \rangle + W(u_2, u_1),$$
$$w(u_1) = w(u_2) + \langle w'(u_2), u_1 - u_2 \rangle + W(u_2, u_1).$$

Using these relations and (2.5.6), we conclude that

$$\phi(u_1) - \phi(u_2) - \langle \phi'(u_2), u_1 - u_2 \rangle \geq (\mu_0 + \mu_1 + \mu_2) W(u_2, u_1) \tag{2.5.8}$$

for any $u_1, u_2 \in Y$, which together with the fact that $\mu_0 + \mu_1 + \mu_2 \geq 0$ then imply that ϕ is convex. Since u^* is an optimal solution of (2.5.7), we have $\langle \phi'(u^*), u - u^* \rangle \geq 0$. Combining this inequality with (2.5.8), we conclude that

$$\phi(u) - \phi(u^*) \geq (\mu_0 + \mu_1 + \mu_2) W(u^*, u),$$

from which the result immediately follows. ■

The following simple result provides a few identities related to y^t and $\tilde{y}^t$ that will be useful for the analysis of the RPDG algorithm.

Lemma 2.5.17 *Let y^t, $\tilde{y}^t$, and $\hat{y}^t$ be defined in (2.3.8), (2.3.9), and (2.3.11), respectively. Then we have, for any $i = 1, \ldots, m$ and $t = 1, \ldots, k$,*

$$\mathbb{E}_t[D_i(y_i^{t-1}, y_i^t)] = p_i D_i(y_i^{t-1}, \hat{y}_i^t), \tag{2.5.9}$$
$$\mathbb{E}_t[D_i(y_i^t, y_i)] = p_i D_i(\hat{y}_i^t, y_i) + (1 - p_i) D_i(y_i^{t-1}, y_i), \tag{2.5.10}$$

for any $y \in \mathcal{Y}$, where $\mathbb{E}_t$ denotes the conditional expectation w.r.t. i_t given $i_1, \ldots, i_{t-1}$.

Proof. (2.5.9) follows immediately from the facts that $\text{Prob}_t\{y_i^t = \hat{y}_i^t\} = \text{Prob}_t\{i_t = i\} = p_i$ and $\text{Prob}_t\{y_i^t = y_i^{t-1}\} = 1 - p_i$. Here Prob_t denotes the conditional probability

w.r.t. i_t given $i_1, \ldots, i_{t-1}$. Similarly, we can show (2.5.10). ■

We now prove an important recursion about the RPDG method.

Lemma 2.5.18 *Let the gap function Q be defined in (2.3.3). Also let x^t and $\hat{y}^t$ be defined in (2.3.10) and (2.3.11), respectively. Then for any $t \geq 1$, we have*

$$\begin{aligned}\mathbb{E}[Q((x^t, \hat{y}^t), z)] \leq\ & \mathbb{E}\left[\eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t)\right] \\ & + \textstyle\sum_{i=1}^m \mathbb{E}\left[\left(p_i^{-1}(1+\tau_t) - 1\right) D_i(y_i^{t-1}, y_i) - p_i^{-1}(1+\tau_t) D_i(y_i^t, y_i)\right] \\ & + \mathbb{E}\left[\langle \tilde{x}^t - x^t, U(\tilde{y}^t - y)\rangle - \tau_t p_{i_t}^{-1} D_{i_t}(y_{i_t}^{t-1}, y_{i_t}^t)\right], \quad \forall z \in Z. \end{aligned} \tag{2.5.11}$$

Proof. It follows from Lemma 2.5.16 applied to (2.3.10) that $\forall x \in X$,

$$\langle x^t - x, U\tilde{y}^t\rangle + h(x^t) + \mu\omega(x^t) - h(x) - \mu\omega(x) \leq \eta_t P(x^{t-1}, x) - (\mu+\eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t). \tag{2.5.12}$$

Moreover, by Lemma 2.5.16 applied to (2.3.11), we have, for any $i = 1, \ldots, m$ and $t = 1, \ldots, k$,

$$\langle -\tilde{x}^t, \hat{y}_i^t - y_i\rangle + J_i(\hat{y}_i^t) - J_i(y_i) \leq \tau_t D_i(y_i^{t-1}, y_i) - (1+\tau_t) D_i(\hat{y}_i^t, y_i) - \tau_t D_i(y_i^{t-1}, \hat{y}_i^t).$$

Summing up these inequalities over $i = 1, \ldots, m$, we have, $\forall y \in \mathcal{Y}$,

$$\langle -\tilde{x}^t, U(\hat{y}^t - y)\rangle + J(\hat{y}^t) - J(y) \leq \textstyle\sum_{i=1}^m \left[\tau_t D_i(y_i^{t-1}, y_i) - (1+\tau_t) D_i(\hat{y}_i^t, y_i) - \tau_t D_i(y_i^{t-1}, \hat{y}_i^t)\right]. \tag{2.5.13}$$

Using the definition of Q in (2.3.3), (2.5.12), and (2.5.13), we have

$$\begin{aligned} Q((x^t, \hat{y}^t), z) \leq\ & \eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t) \\ & + \textstyle\sum_{i=1}^m \left[\tau_t D_i(y_i^{t-1}, y_i) - (1+\tau_t) D_i(\hat{y}_i^t, y_i) - \tau_t D_i(y_i^{t-1}, \hat{y}_i^t)\right] \\ & + \langle \tilde{x}^t, U(\hat{y}^t - y)\rangle - \langle x^t, U(\tilde{y}^t - y)\rangle + \langle x, U(\tilde{y}^t - \hat{y}^t)\rangle. \end{aligned} \tag{2.5.14}$$

Also observe that by (2.3.8), (2.3.12), (2.5.9), and (2.5.10),

$$\begin{aligned} D_i(y_i^{t-1}, y_i^t) &= 0,\ \forall i \neq i_t, \\ \mathbb{E}[\langle x, U(\tilde{y}^t - \hat{y}^t)\rangle] &= 0, \\ \mathbb{E}[\langle \tilde{x}^t, U\hat{y}^t\rangle] &= \mathbb{E}[\langle \tilde{x}^t, U\tilde{y}^t\rangle], \\ \mathbb{E}[D_i(y_i^{t-1}, \hat{y}_i^t)] &= \mathbb{E}[p_i^{-1} D_i(y_i^{t-1}, y_i^t)] \\ \mathbb{E}[D_i(\hat{y}_i^t, y_i)] &= p_i^{-1}\mathbb{E}[D_i(y_i^t, y_i)] - (p_i^{-1} - 1)\mathbb{E}[D_i(y_i^{t-1}, y_i)], \end{aligned}$$

Taking expectation on both sides of (2.5.14) and using the above observations, we obtain (2.5.11). ■

We are now ready to establish a general convergence result which holds for both PDG and RPDG.

Proposition 2.5.19 *Suppose that* $\{\tau_t\}$, $\{\eta_t\}$, *and* $\{\alpha_t\}$ *in the RPDG method satisfy*

$$\theta_t \left(p_i^{-1}(1+\tau_t) - 1\right) \leq p_i^{-1}\theta_{t-1}(1+\tau_{t-1}), i = 1, \ldots, m; t = 2, \ldots, k, \tag{2.5.15}$$

$$\theta_t \eta_t \leq \theta_{t-1}(\mu + \eta_{t-1}), t = 2, \ldots, k, \tag{2.5.16}$$

$$\tfrac{\eta_k}{4} \geq \tfrac{L_i(1-p_i)^2}{m\tau_k p_i}, i = 1, \ldots, m, \tag{2.5.17}$$

$$\tfrac{\eta_{t-1}}{2} \geq \tfrac{L_i\alpha_t}{m\tau_t p_i} + \tfrac{(1-p_j)^2 L_j}{m\tau_{t-1}p_j},\ i, j \in \{1, \ldots, m\}; t = 2, \ldots, k, \tag{2.5.18}$$

$$\tfrac{\eta_k}{2} \geq \tfrac{\sum_{i=1}^m (p_i L_i)}{m(1+\tau_k)}, \tag{2.5.19}$$

$$\alpha_t \theta_t = \theta_{t-1}, t = 2, \ldots, k, \tag{2.5.20}$$

for some $\theta_t \geq 0$, $t = 1, \ldots, k$. *Then, for any* $k \geq 1$ *and any given* $z \in Z$, *we have*

$$\begin{aligned} \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[Q((x^t, \hat{y}^t), z)] \leq\ & \eta_1\theta_1 P(x^0, x) - (\mu + \eta_k)\theta_k \mathbb{E}[P(x^k, x)] \\ & + \textstyle\sum_{i=1}^m \theta_1 \left(p_i^{-1}(1+\tau_1) - 1\right) D_i(y_i^0, y_i). \end{aligned} \tag{2.5.21}$$

Proof. Multiplying both sides of (2.5.11) by θ_t and summing the resulting inequalities, we have

$$\begin{aligned}\mathbb{E}[\textstyle\sum_{t=1}^k \theta_t Q((x^t,\hat{y}^t),z)] &\le \mathbb{E}\left[\textstyle\sum_{t=1}^k \theta_t \left(\eta_t P(x^{t-1},x) - (\mu+\eta_t)P(x^t,x) - \eta_t P(x^{t-1},x^t)\right)\right] \\ &+ \textstyle\sum_{i=1}^m \mathbb{E}\left\{\sum_{t=1}^k \theta_t \left[\left(p_i^{-1}(1+\tau_t)-1\right) D_i(y_i^{t-1},y_i) - p_i^{-1}(1+\tau_t) D_i(y_i^t,y_i)\right]\right\} \\ &+ \mathbb{E}\left[\textstyle\sum_{t=1}^k \theta_t \left(\langle \tilde{x}^t - x^t, U(\tilde{y}^t - y)\rangle - \tau_t p_{i_t}^{-1} D_{i_t}(y_{i_t}^{t-1}, y_{i_t}^t)\right)\right],\end{aligned}$$

which, in view of the assumptions in (2.5.16) and (2.5.15), then implies that

$$\begin{aligned}\mathbb{E}[\textstyle\sum_{t=1}^k \theta_t Q((x^t,\hat{y}^t),z)] &\le \eta_1\theta_1 P(x^0,x) - (\mu+\eta_k)\theta_k \mathbb{E}[P(x^k,x)] \\ &+ \textstyle\sum_{i=1}^m \mathbb{E}\left[\theta_1\left(p_i^{-1}(1+\tau_1)-1\right) D_i(y_i^0,y_i) - p_i^{-1}\theta_k(1+\tau_k) D_i(y_i^k,y_i)\right] \\ &- \mathbb{E}\left[\textstyle\sum_{t=1}^k \theta_t \Delta_t\right], \qquad (2.5.22)\end{aligned}$$

where

$$\Delta_t := \eta_t P(x^{t-1},x^t) - \langle \tilde{x}^t - x^t, U(\tilde{y}^t - y)\rangle + \tau_t p_{i_t}^{-1} D_{i_t}(y_{i_t}^{t-1}, y_{i_t}^t). \qquad (2.5.23)$$

We now provide a bound on $\sum_{t=1}^k \theta_t \Delta_t$ in (2.5.22). Note that by (2.3.7), we have

$$\begin{aligned}\langle \tilde{x}^t - x^t, U(\tilde{y}^t - y)\rangle &= \langle x^{t-1} - x^t, U(\tilde{y}^t - y)\rangle - \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^t - y)\rangle \\ &= \langle x^{t-1} - x^t, U(\tilde{y}^t - y)\rangle - \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^{t-1} - y)\rangle \\ &\quad - \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^t - \tilde{y}^{t-1})\rangle \\ &= \langle x^{t-1} - x^t, U(\tilde{y}^t - y)\rangle - \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^{t-1} - y)\rangle \\ &\quad - \alpha_t p_{i_t}^{-1} \langle x^{t-2} - x^{t-1}, y_{i_t}^t - y_{i_t}^{t-1}\rangle \\ &\quad - \alpha_t (p_{i_{t-1}}^{-1} - 1)\langle x^{t-2} - x^{t-1}, y_{i_{t-1}}^{t-2} - y_{i_{t-1}}^{t-1}\rangle, \qquad (2.5.24)\end{aligned}$$

where the last identity follows from the observation that by (2.3.8) and (2.3.9),

$$\begin{aligned} U(\tilde{y}^t - \tilde{y}^{t-1}) &= \textstyle\sum_{i=1}^m \left\{ \left[p_i^{-1}(y_i^t - y_i^{t-1}) + y_i^{t-1} \right] - \left[p_i^{-1}(y_i^{t-1} - y_i^{t-2}) + y_i^{t-2} \right] \right\} \\ &= \textstyle\sum_{i=1}^m \left\{ \left[p_i^{-1} y_i^t - (p_i^{-1} - 1) y_i^{t-1} \right] - \left[p_i^{-1} y_i^{t-1} - (p_i^{-1} - 1) y_i^{t-2} \right] \right\} \\ &= \textstyle\sum_{i=1}^m \left[p_i^{-1}(y_i^t - y_i^{t-1}) + (p_i^{-1} - 1)(y_i^{t-2} - y_i^{t-1}) \right] \\ &= p_{i_t}^{-1}(y_{i_t}^t - y_{i_t}^{t-1}) + (p_{i_{t-1}}^{-1} - 1)(y_{i_{t-1}}^{t-2} - y_{i_{t-1}}^{t-1}). \end{aligned}$$

Using relation (2.5.24) in the definition of Δ_t in (2.5.23), we have

$$\begin{aligned} \textstyle\sum_{t=1}^k \theta_t \Delta_t = \textstyle\sum_{t=1}^k \theta_t \big[& \eta_t P(x^{t-1}, x^t) \\ & - \langle x^{t-1} - x^t, U(\tilde{y}^t - y) \rangle + \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^{t-1} - y) \rangle \\ & + \alpha_t p_{i_t}^{-1} \langle x^{t-2} - x^{t-1}, y_{i_t}^t - y_{i_t}^{t-1} \rangle + \alpha_t (p_{i_{t-1}}^{-1} - 1) \langle x^{t-2} - x^{t-1}, y_{i_{t-1}}^{t-2} - y_{i_{t-1}}^{t-1} \rangle \\ & + p_{i_t}^{-1} \tau_t D_{i_t}(y_{i_t}^{t-1}, y_{i_t}^t) \big]. \end{aligned} \tag{2.5.25}$$

Observe that by (2.5.20) and the fact that $x^{-1} = x^0$,

$$\begin{aligned} & \textstyle\sum_{t=1}^k \theta_t \left[\langle x^{t-1} - x^t, U(\tilde{y}^t - y) \rangle - \alpha_t \langle x^{t-2} - x^{t-1}, U(\tilde{y}^{t-1} - y) \rangle \right] \\ &= \theta_k \langle x^{k-1} - x^k, U(\tilde{y}^k - y) \rangle \\ &= \theta_k \langle x^{k-1} - x^k, U(y^k - y) \rangle + \theta_k \langle x^{k-1} - x^k, U(\tilde{y}^k - y^k) \rangle \\ &= \theta_k \langle x^{k-1} - x^k, U(y^k - y) \rangle + \theta_k (p_{i_k}^{-1} - 1) \langle x^{k-1} - x^k, y_{i_k}^k - y_{i_k}^{k-1} \rangle, \end{aligned}$$

where the last identity follows from the definitions of y^k and $\tilde{y}^k$ in (2.3.8) and (2.3.9), respectively. Also, by the strong convexity of P and D_i, we have

$$P(x^{t-1}, x^t) \geq \tfrac{1}{2} \|x^{t-1} - x^t\|^2 \quad \text{and} \quad D_{i_t}(y_{i_t}^{t-1}, y_{i_t}^t) \geq \tfrac{m}{2L_{i_t}} \|y_{i_t}^{t-1} - y_{i_t}^t\|^2.$$

Using the previous three relations in (2.5.25), we have

$$\begin{aligned}\sum_{t=1}^k \theta_t \Delta_t \geq \sum_{t=1}^k \theta_t \Big[& \tfrac{\eta_t}{2}\|x^{t-1}-x^t\|^2 + \alpha_t p_{i_t}^{-1}\langle x^{t-2}-x^{t-1}, y_{i_t}^t - y_{i_t}^{t-1}\rangle \\ & + \alpha_t(p_{i_{t-1}}^{-1}-1)\langle x^{t-2}-x^{t-1}, y_{i_{t-1}}^{t-2}-y_{i_{t-1}}^{t-1}\rangle + \tfrac{m\tau_t}{2L_{i_t}p_{i_t}}\|y_{i_t}^{t-1}-y_{i_t}^t\|^2\Big] \\ & - \theta_k\langle x^{k-1}-x^k, U(y^k-y)\rangle - \theta_k(p_{i_k}^{-1}-1)\langle x^{k-1}-x^k, y_{i_k}^k - y_{i_k}^{k-1}\rangle.\end{aligned}$$

Regrouping the terms in the above relation, and the fact that $x^{-1}=x^0$, we obtain

$$\begin{aligned}\sum_{t=1}^k \theta_t\Delta_t &\geq \theta_k\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle\right] \\ &\quad + \theta_k\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - (p_{i_k}^{-1}-1)\langle x^{k-1}-x^k, y_{i_k}^k - y_{i_k}^{k-1}\rangle + \tfrac{m\tau_k}{4L_{i_k}p_{i_k}}\|y_{i_k}^{k-1}-y_{i_k}^k\|^2\right] \\ &\quad + \sum_{t=2}^k \theta_t\left[\tfrac{\alpha_t}{p_{i_t}}\langle x^{t-2}-x^{t-1}, y_{i_t}^t - y_{i_t}^{t-1}\rangle + \tfrac{m\tau_t}{4L_{i_t}p_{i_t}}\|y_{i_t}^{t-1}-y_{i_t}^t\|^2\right] \\ &\quad + \sum_{t=2}^k \left[\alpha_t\theta_t(p_{i_{t-1}}^{-1}-1)\langle x^{t-2}-x^{t-1}, y_{i_{t-1}}^{t-2}-y_{i_{t-1}}^{t-1}\rangle + \tfrac{m\tau_{t-1}\theta_{t-1}}{4L_{i_{t-1}}p_{i_{t-1}}}\|y_{i_{t-1}}^{t-2}-y_{i_{t-1}}^{t-1}\|^2\right] \\ &\quad + \sum_{t=2}^k \tfrac{\theta_{t-1}\eta_{t-1}}{2}\|x^{t-2}-x^{t-1}\|^2 \\ &\geq \theta_k\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle\right] \\ &\quad + \theta_k\left(\tfrac{\eta_k}{4} - \tfrac{L_{i_k}(1-p_{i_k})^2}{m\tau_k p_{i_k}}\right)\|x^{k-1}-x^k\|^2 \\ &\quad + \sum_{t=2}^k\left[\tfrac{\theta_{t-1}\eta_{t-1}}{2} - \tfrac{L_{i_t}\alpha_t^2\theta_t}{m\tau_t p_{i_t}} - \tfrac{\alpha_t^2\theta_t^2(1-p_{i_{t-1}})^2 L_{i_{t-1}}}{m\tau_{t-1}\theta_{t-1}p_{i_{t-1}}}\right]\|x^{t-2}-x^{t-1}\|^2 \\ &= \theta_k\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle\right] \\ &\quad + \theta_k\left(\tfrac{\eta_k}{4} - \tfrac{L_{i_k}(1-p_{i_k})^2}{m\tau_k p_{i_k}}\right)\|x^{k-1}-x^k\|^2 \\ &\quad + \sum_{t=2}^k\theta_{t-1}\left(\tfrac{\eta_{t-1}}{2} - \tfrac{L_{i_t}\alpha_t}{m\tau_t p_{i_t}} - \tfrac{(1-p_{i_{t-1}})^2 L_{i_{t-1}}}{m\tau_{t-1}p_{i_{t-1}}}\right)\|x^{t-2}-x^{t-1}\|^2 \\ &\geq \theta_k\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle\right], \end{aligned} \tag{2.5.26}$$

where the second inequality follows from the simple relation that

$$b\langle u,v\rangle + a\|v\|^2/2 \geq -b^2\|u\|^2/(2a), \forall a>0, \tag{2.5.27}$$

and the last inequality follows from (2.5.17) and (2.5.18). Plugging the bound (2.5.26) into

(2.5.22), we have

$$\sum_{t=1}^{k}\theta_t\mathbb{E}[Q((x^t,\hat{y}^t),z)] \le \theta_1\eta_1 P(x^0,x) - \theta_k(\mu+\eta_k)\mathbb{E}[P(x^k,x)] + \sum_{i=1}^{m}\theta_1\left(p_i^{-1}(1+\tau_1)-1\right)D_i(y_i^0,y_i)$$
$$- \theta_k\mathbb{E}\left[\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle + \sum_{i=1}^{m}p_i^{-1}(1+\tau_k)D_i(y_i^k,y_i)\right].$$

Also observe that by (2.5.19) and (2.5.27),

$$\begin{aligned}
&\tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 - \langle x^{k-1}-x^k, U(y^k-y)\rangle \\
&+ \textstyle\sum_{i=1}^{m}p_i^{-1}(1+\tau_k)D_i(y_i^k,y_i) \\
&\ge \tfrac{\eta_k}{4}\|x^{k-1}-x^k\|^2 + \textstyle\sum_{i=1}^{m}\left[-\langle x^{k-1}-x^k, y_i^k-y_i\rangle + \tfrac{m(1+\tau_k)}{2L_ip_i}\|y_i^k-y_i\|^2\right] \\
&\ge \left(\tfrac{\eta_k}{4} - \tfrac{\sum_{i=1}^{m}(p_iL_i)}{2m(1+\tau_k)}\right)\|x^{k-1}-x^k\|^2 \ge 0,
\end{aligned}$$

The result then immediately follows by combining the above two conclusion. ■

2.5.3 Proof of Main Convergence Results

We now provide a proof for Theorem 2.2.3 which describes the main convergence properties of the deterministic PDG method.

We first specialize Proposition 2.5.19 for the PDG method applied to (2.2.7).

Proposition 2.5.20 *Suppose that $\{\tau_t\}$, $\{\eta_t\}$, and $\{\alpha_t\}$ in the PDG method satisfy*

$$\theta_t\tau_t \le \theta_{t-1}(1+\tau_{t-1}), t=2,\ldots,k, \tag{2.5.28}$$

$$\theta_t\eta_t \le \theta_{t-1}(\mu+\eta_{t-1}), t=2,\ldots,k, \tag{2.5.29}$$

$$\eta_{t-1}\tau_t \ge 2L_f\alpha_t, t=2,\ldots,k, \tag{2.5.30}$$

$$\eta_k(1+\tau_k) \ge 2L_f, \tag{2.5.31}$$

$$\alpha_t = \theta_{t-1}/\theta_t, t=2,\ldots,k, \tag{2.5.32}$$

for some $\theta_t \geq 0$, $t = 1, \ldots, k$. Also let us denote $z^t = (x^t, g^t)$, and

$$\bar{z}^k := \left(\sum_{t=1}^k \theta_t\right)^{-1} \sum_{t=1}^k \theta_t z^t. \tag{2.5.33}$$

Then, for any $k \geq 1$ and any given $(x, g) \in X \times \mathcal{G}$, we have

$$\left(\sum_{t=1}^k \theta_t\right) Q_f(\bar{z}^k, z) + \theta_k(\mu + \eta_k) P(x^k, x) \leq \theta_1 \eta_1 P(x^0, x) + \theta_1 \tau_1 D_f(g^0, g). \tag{2.5.34}$$

Proof. Notice that in the deterministic PDG method, we have $m = 1$, $p_i = 1$, and $\hat{y}^t = g^t$. It can be easily seen that the assumptions in (2.5.15)-(2.5.20) are implied by those in (2.5.28)-(2.5.32). It then follows from (2.5.21) that

$$\sum_{t=1}^k \theta_t Q_f(z^t, z) \leq \theta_1 \eta_1 P(x^0, x) - \theta_k(\mu + \eta_k) P(x^k, x) + \theta_1 \tau_1 D_f(g^0, g).$$

Dividing both sides of the above inequality by $\sum_{t=1}^k \theta_t$ and using the convexity of $Q(\bar{z}, z)$ w.r.t. $\bar{z}$, we have

$$\left(\sum_{t=1}^k \theta_t\right) Q_f(\bar{z}^k, z) \leq \sum_{t=1}^k \theta_t Q_f(z^t, z) \leq \theta_1 \eta_1 P(x^0, x) - \theta_k(\mu + \eta_k) P(x^k, x) + \theta_1 \tau_1 D_f(g^0, g).$$

Rearranging the terms in the above relation, we obtain (2.5.34). ∎

We are now ready to show Theorem 2.2.3.

Proof of Theorem 2.2.3 We first show part a). It can be easily checked that (2.5.28)-(2.5.32) are satisfied with the selection of $\{\tau_t\}$, $\{\eta_t\}$, $\{\alpha_t\}$, and $\{\theta_t\}$ in (2.2.25). Using (2.5.34) (with $x = x^*$ and $y = y^*$), (2.5.3), and the fact that $Q_f(\bar{z}, z^*) \geq 0$, we have

$$\theta_k(\mu + \eta_k) P(x^k, x^*) \leq \theta_1(\eta_1 + L_f \tau_1) P(x^0, x^*), \quad \forall k \geq 1.$$

Using the parameter settings in (2.2.25), we conclude that

$$P(x^k, x^*) \leq \frac{\theta_1(\eta_1+L_f\tau_1)}{\theta_k(\mu+\eta_k)}P(x^0, x^*) = \frac{(\sqrt{2L_f\mu}+L_f\sqrt{2L_f/\mu})}{\alpha(\mu+\sqrt{2L_f\mu})}\alpha^k P(x^0, x^*) = \frac{\mu+L_f}{\mu}\alpha^k P(x^0, x^*).$$

Also using (2.5.34) and the fact that $P(x^k, x) \geq 0$, we have

$$\left(\sum_{t=1}^k \theta_t\right) Q_f(\bar{z}^k, z) \leq \theta_1\eta_1 P(x^0, x) + \theta_1\tau_1 D_f(g^0, g), \quad \forall z \in Z. \tag{2.5.35}$$

Denoting $\bar{g}_*^k := \nabla f(\bar{x}^k)$, we conclude from (2.5.3) that

$$\begin{aligned} D_f(g^0, \bar{g}_*^k) &\leq \tfrac{L_f}{2}\|\bar{x}^k - x^0\|^2 \leq \tfrac{L_f}{2}[\textstyle\sum_{t=1}^k \theta_t]^{-1}\sum_{t=1}^k \theta_t\|x^t - x^0\|^2 \\ &\leq \tfrac{L_f}{2}[\textstyle\sum_{t=1}^k \theta_t]^{-1}\sum_{t=1}^k \theta_t(\|x^t - x^*\|^2 + \|x^0 - x^*\|^2) \\ &\leq \tfrac{L_f}{2}\left[\tfrac{2(\mu+L_f)}{\mu}P(x^0, x^*) + \|x^0 - x^*\|^2\right] \leq L_f\left(\tfrac{2\mu+L_f}{\mu}\right)P(x^0, x^*), \end{aligned}$$

where the second inequality follows from the convexity of $\|\cdot\|^2$, the third inequality follows from the triangular inequality, the fourth inequality follows from $\|x^t - x^*\|^2 \leq 2P(x^t, x^*)$ and (2.2.26), and the last inequality follows from $\|x^0 - x^*\|^2 \leq 2P(x^0, x^*)$. Also note that by the definition of θ_t, we have

$$\textstyle\sum_{t=1}^k \theta_t = \sum_{t=1}^k \alpha^{-t} = \frac{1-\alpha^k}{(1-\alpha)\alpha^k} \geq \frac{1}{\alpha^k}, \tag{2.5.36}$$

where the last inequality follows from the fact that $\alpha \leq 1$ due to (2.2.25). Fixing $x = x^*$, $g = \bar{g}_*^k$ in (2.5.35) and using the above two relations, we obtain

$$\begin{aligned} Q_f(\bar{z}^k, (x^*, \bar{g}_*^k)) &\leq \alpha^k\left[\theta_1\eta_1 P(x^0, x^*) + L_f\theta_1\tau_1\left(\tfrac{2\mu+L_f}{\mu}\right)P(x^0, x^*)\right] \\ &\leq (\mu + \sqrt{2L_f\mu})\alpha^k\left[P(x^0, x^*) + \tfrac{L_f}{\mu}(2 + \tfrac{L_f}{\mu})P(x^0, x^*)\right] \\ &= \tfrac{\mu\alpha^k}{1-\alpha}\left[P(x^0, x^*) + \tfrac{L_f}{\mu}(2 + \tfrac{L_f}{\mu})P(x^0, x^*)\right]. \end{aligned}$$

The result in (2.2.27) then directly follows from the above relation and (2.2.22). If X is bounded, the result in (2.2.28) then follows from the above relation, (2.2.22), and (2.2.23).

We now show part b). It is trivial to check that the conditions in (2.5.28)-(2.5.32) hold by using our selection of $\{\tau_t\}$, $\{\eta_t\}$, $\{\alpha_t\}$, and $\{\theta_t\}$. Using (2.5.34) and the facts $\tau_1 = 0$ and $P(x^k, x) \geq 0$, we have

$$\left(\sum_{t=1}^{k} \theta_t\right) Q_f(\bar{z}^k, z) \leq \theta_1 \eta_1 P(x^0, x) = 4L_f P(x^0, x).$$

which, in view of (2.2.21) and (2.2.22) and the fact that $\sum_{t=1}^{k} \theta_t = k(k+1)/2$, clearly implies (2.2.30). In case X is bounded, the result in (2.2.31) immediately follows from (2.2.22), (2.2.23), and the above inequality. ■

We are now ready to provide a proof for Theorem 2.3.4, which describes the main convergence properties of the RPDG method applied to strongly convex problems with $\mu > 0$.

Proof of Theorem 2.3.4. It can be easily checked that the conditions in (2.5.15)-(2.5.20) are satisfied with our requirements (2.3.19)-(2.3.22) of $\{\tau_t\}$, $\{\eta_t\}$, $\{\alpha_t\}$, and $\{\theta_t\}$. Using the fact that $Q((x^t, \hat{y}^t), z^*) \geq 0$, we then conclude from (2.5.21) (with $x = x^*$ and $y = y^*$) that, for any $k \geq 1$,

$$\mathbb{E}[P(x^k, x^*)] \leq \tfrac{1}{\theta_k(\mu+\eta)} \left[\theta_1 \eta P(x^0, x^*) + \tfrac{\theta_1 \alpha}{1-\alpha} D(y^0, y^*)\right] \leq \left(1 + \tfrac{L_f \alpha}{(1-\alpha)\eta}\right) \alpha^k P(x^0, x^*),$$

where the first inequality follows from (2.3.19) and (2.3.20), and the second inequality follows from (2.3.21) and (2.5.1).

Let us denote $\bar{y}^k \equiv (\sum_{t=1}^{k} \theta_t)^{-1} \sum_{t=1}^{k} (\theta_t \hat{y}^t)$, $\bar{z}^k = (\bar{x}^k, \bar{y}^k)$. In view of (2.5.4), the convexity of $\|\cdot\|$, and (2.2.2), we have

$$\mathbb{E}[\Psi(\bar{x}^k) - \Psi(x^*)] \leq \mathbb{E}[Q(\bar{z}^k, z^*)] + \tfrac{L_f}{2} (\textstyle\sum_{t=1}^{k} \theta_t)^{-1} \mathbb{E}[\sum_{t=1}^{k} \theta_t \|x^t - x^*\|^2]$$

$$\leq \mathbb{E}[Q(\bar{z}^k, z^*)] + L_f(\textstyle\sum_{t=1}^k \theta_t)^{-1}\mathbb{E}[\sum_{t=1}^k \theta_t P(x^t, x^*)]. \qquad (2.5.37)$$

Using (2.5.21) (with $x = x^*$ and $y = y^*$), the fact that $P(x^k, x^*) \geq 0$, and (2.5.36), we obtain

$$\mathbb{E}[Q(\bar{z}^k, z^*)] \leq \left(\textstyle\sum_{t=1}^k \theta_t\right)^{-1} \sum_{t=1}^k \theta_t \mathbb{E}[Q((x^t, \hat{y}^t), z^*)] \leq \alpha^k \left(\alpha^{-1}\eta + \tfrac{L_f}{1-\alpha}\right) P(x^0, x^*).$$

We conclude from (2.3.23) and the definition of $\{\theta_t\}$ that

$$\begin{aligned}
(\textstyle\sum_{t=1}^k \theta_t)^{-1}\mathbb{E}[\sum_{t=1}^k \theta_t P(x^t, x^*)] &= (\textstyle\sum_{t=1}^k \alpha^{-t})^{-1}\sum_{t=1}^k \alpha^{-t}(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta})\alpha^t P(x^0, x^*) \\
&\leq \tfrac{1-\alpha}{\alpha^{-k}-1}\textstyle\sum_{t=1}^k \tfrac{\alpha^t}{\alpha^{3t/2}}(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta})P(x^0, x^*) \\
&= \tfrac{1-\alpha}{\alpha^{-k}-1}\tfrac{\alpha^{-k/2}-1}{1-\alpha^{1/2}}(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta})P(x^0, x^*) \\
&= \tfrac{1+\alpha^{1/2}}{1+\alpha^{-k/2}}(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta})P(x^0, x^*) \leq 2\alpha^{k/2}(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta})P(x^0, x^*).
\end{aligned}$$

Using the above two relations, and (2.5.37), we obtain

$$\begin{aligned}
\mathbb{E}[\Psi(\bar{x}^k) - \Psi(x^*)] &\leq \alpha^k \left(\alpha^{-1}\eta + \tfrac{L_f}{1-\alpha}\right) P(x^0, x^*) + L_f 2\alpha^{k/2} \left(1 + \tfrac{L_f\alpha}{(1-\alpha)\eta}\right) P(x^0, x^*) \\
&\leq \alpha^{k/2} \left(\alpha^{-1}\eta + \tfrac{3-2\alpha}{1-\alpha}L_f + \tfrac{2L_f^2\alpha}{(1-\alpha)\eta}\right) P(x^0, x^*).
\end{aligned}$$

■

2.5.4 Proof of the Lower Complexity Bound

This subsection is devoted to the proof of Theorem 2.3.7, which describes the performance limit for randomized incremental gradient methods.

The following result provides an explicit expression for the optimal solution of (2.3.37).

Lemma 2.5.21 *Let q be defined in (2.3.42), $x_{i,j}^*$ is the j-th element of x_i, and define*

$$x_{i,j}^* = q^j, i = 1, \ldots, m; j = 1, \ldots, \tilde{n}. \qquad (2.5.38)$$

Then x^* is the unique optimal solution of (2.3.37).

Proof. It can be easily seen that q is the smallest root of the equation

$$q^2 - 2\tfrac{\mathcal{Q}+1}{\mathcal{Q}-1}q + 1 = 0. \tag{2.5.39}$$

Note that x^* satisfies the optimality condition of (2.3.37), i.e.,

$$\left(A + \tfrac{4}{\mathcal{Q}-1}I\right) x_i^* = e_1, \quad i = 1, \ldots, m. \tag{2.5.40}$$

Indeed, we can write the coordinate form of (2.5.40) as

$$2\tfrac{\mathcal{Q}+1}{\mathcal{Q}-1}x_{i,1}^* - x_{i,2}^* = 1, \tag{2.5.41}$$

$$x_{i,j+1}^* - 2\tfrac{\mathcal{Q}+1}{\mathcal{Q}-1}x_{i,j}^* + x_{i,j-1}^* = 0,\ j = 2, 3, \ldots, \tilde{n} - 1, \tag{2.5.42}$$

$$-(\kappa + \tfrac{4}{\mathcal{Q}-1})x_{i,\tilde{n}}^* + x_{i,\tilde{n}-1}^* = 0, \tag{2.5.43}$$

where the first two equations follow directly from the definition of x^* and relation (2.5.39), and the last equation is implied by the definitions of κ and x^* in (2.3.39) and (2.5.38), respectively. ■

We also need a few technical results to establish the lower complexity bounds.

Lemma 2.5.22 *a) For any $x > 1$, we have*

$$\log(1 - \tfrac{1}{x}) \geq -\tfrac{1}{x-1}. \tag{2.5.44}$$

b) Let $\rho, q, \bar{q} \in (0, 1)$ be given. If we have

$$\tilde{n} \geq \tfrac{t \log \bar{q} + \log(1-\rho)}{2 \log q},$$

for any $t \geq 0$, then

$$\bar{q}^t - q^{2\tilde{n}} \geq \rho \bar{q}^t (1 - q^{2\tilde{n}}).$$

Proof. We first show part a). Denote $\phi(x) = \log(1 - \frac{1}{x}) + \frac{1}{x-1}$. It can be easily seen that $\lim_{x \to +\infty} \phi(x) = 0$. Moreover, for any $x > 1$, we have

$$\phi'(x) = \frac{1}{x(x-1)} - \frac{1}{(x-1)^2} = \frac{1}{x-1}\left(\frac{1}{x} - \frac{1}{x-1}\right) < 0,$$

which implies that ϕ is a strictly decreasing function for $x > 1$. Hence, we must have $\phi(x) > 0$ for any $x > 1$. Part b) follows from the following simple calculation.

$$\bar{q}^t - q^{2\tilde{n}} - \rho \bar{q}^t (1 - q^{2\tilde{n}}) = (1 - \rho)\bar{q}^t - q^{2\tilde{n}} + \rho \bar{q}^t q^{2\tilde{n}} \geq (1 - \rho)\bar{q}^t - q^{2\tilde{n}} \geq 0.$$

■

We are now ready to prove Theorem 2.3.7.

Proof of Theorem 2.3.7 Without loss of generality, we may assume that the initial point $x_i^0 = 0$, $i = 1, \ldots, m$. Indeed, the incremental gradient methods described in Subsection 3.3 are invariant with respect to a simultaneous shift of the decision variables. In other words, the sequence of iterates $\{x^k\}$, which is generated by such a method for minimizing the function $\Psi(x)$ starting from x^0, is just a shift of the sequence generated for minimizing $\bar{\Psi}(x) = \Psi(x + x^0)$ starting from the origin.

Now let k_i, $i = 1, \ldots, m$, denote the number of times that the gradients of the component function f_i are computed from iteration 1 to k. Clearly k_i's are binomial random variables supported on $\{0, 1, \ldots, k\}$ such that $\sum_{i=1}^m k_i = k$. Also observe that we must have $x_{i,j}^k = 0$ for any $k \geq 0$ and $k_i + 1 \leq j \leq \tilde{n}$, because each time the gradient ∇f_i is computed, the incremental gradient methods add at most one more nonzero entry to the

i-th component of x^k due to the structure of the gradient ∇f_i. Therefore, we have

$$\frac{\|x^k-x^*\|_2^2}{\|x^0-x^*\|_2^2} = \frac{\sum_{i=1}^m \|x_i^k - x_i^*\|_2^2}{\sum_{i=1}^m \|x_i^*\|^2} \geq \frac{\sum_{i=1}^m \sum_{j=k_i+1}^{\tilde{n}} (x_{i,j}^*)^2}{\sum_{i=1}^m \sum_{j=1}^{\tilde{n}} (x_{i,j}^*)^2} = \frac{\sum_{i=1}^m (q^{2k_i} - q^{2\tilde{n}})}{m(1-q^{2\tilde{n}})}. \quad (2.5.45)$$

Observing that for any $i = 1, \ldots, m$,

$$\mathbb{E}[q^{2k_i}] = \sum_{t=0}^k \left[q^{2t}\binom{k}{t} p_i^t (1-p_i)^{k-t}\right] = [1-(1-q^2)p_i]^k,$$

we then conclude from (2.5.45) that

$$\frac{\mathbb{E}[\|x^k-x^*\|_2^2]}{\|x^0-x^*\|_2^2} \geq \frac{\sum_{i=1}^m [1-(1-q^2)p_i]^k - mq^{2\tilde{n}}}{m(1-q^{2\tilde{n}})}.$$

Noting that $[1-(1-q^2)p_i]^k$ is convex w.r.t. p_i for any $p_i \in [0,1]$ and $k \geq 1$, by minimizing the RHS of the above bound w.r.t. p_i, $i = 1, \ldots, m$, subject to $\sum_{i=1}^m p_i = 1$ and $p_i \geq 0$, we conclude that

$$\frac{\mathbb{E}[\|x^k-x^*\|_2^2]}{\|x^0-x^*\|_2^2} \geq \frac{[1-(1-q^2)/m]^k - q^{2\tilde{n}}}{1-q^{2\tilde{n}}} \geq \frac{1}{2}[1-(1-q^2)/m]^k, \quad (2.5.46)$$

for any possible selection of p_i, $i = 1, \ldots, m$, satisfying (2.3.40) and any

$$n \geq n_0 := \frac{m \log[(1-(1-q^2)/m)^k/2]}{2\log q}. \quad (2.5.47)$$

Here, the last inequality in (2.5.46) follows from Lemma 2.5.22.b). Noting that

$$1-(1-q^2)/m = 1 - \left[1 - \left(\frac{\sqrt{\mathcal{Q}}-1}{\sqrt{\mathcal{Q}}+1}\right)^2\right]\frac{1}{m} = 1 - \frac{1}{m} + \frac{1}{m}\left(1 - \frac{2}{\sqrt{\mathcal{Q}}+1}\right)^2$$
$$= 1 - \frac{4}{m(\sqrt{\mathcal{Q}}+1)} + \frac{4}{m(\sqrt{\mathcal{Q}}+1)^2} = 1 - \frac{4\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2},$$

we then conclude from (2.5.46) and Lemma 2.5.22.a) that

$$\begin{aligned}\frac{\mathbb{E}[\|x^k-x^*\|_2^2]}{\|x^0-x^*\|_2^2} &\geq \frac{1}{2}\left[1-\frac{4\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2}\right]^k = \frac{1}{2}\exp\left(k\log\left(1-\frac{4\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2}\right)\right)\\ &\geq \frac{1}{2}\exp\left(-\frac{4k\sqrt{\mathcal{Q}}}{m(\sqrt{\mathcal{Q}}+1)^2-4\sqrt{\mathcal{Q}}}\right).\end{aligned}$$

Now it suffices to show that n_0 in (2.5.47) is smaller than the simplified bound $\underline{n}(m,k)$ in (2.3.44). Indeed, observing

$$\begin{aligned}-\log[(1-(1-q^2)/m)^k/2] &= -k\log(1-(1-q^2)/m)+\log 2\\ &\leq \frac{k}{m/(1-q^2)-1}+\log 2\\ &= \frac{k(1-q^2)}{m-1+q^2}+\log 2\\ &\leq \frac{2k}{m}+1,\end{aligned}$$

where the first inequality follows from (2.5.44) and the last inequality follows from $m \geq 2$, we have

$$\begin{aligned}n_0 = \frac{m\log[(1-(1-q^2)/m)^k/2]}{2\log q} &= \frac{-m\log[(1-(1-q^2)/m)^k/2]}{2\log(1/q)}\\ &\leq \left(k+\frac{m}{2}\right)\frac{1}{\log(1/q)} = \underline{n}(m,k).\end{aligned}$$

■

2.6 Concluding Remarks of This Chapter

In this chapter, we present a new class of optimal first-order methods, referred to as primal-dual gradient methods, for solving the finite-sum composite convex optimization problems given in the form of (1.1.1). The optimal convergence of this algorithm has been established based on the primal-dual optimality gap for the ergodic mean of iterates, i.e., $\bar{z}^k$, and the distance from the iterate x^k to the optimal solution x^*. We also develop a ran-

domized primal-dual gradient method which needs to compute the gradient of only one randomly selected component f_i. The complexity bounds of the randomized primal-dual gradient method have been established in terms of the distance from the iterate x^k to the optimal solution, and the primal optimality gap based on the ergodic mean of iterates, i.e., $\mathbb{E}[\Psi(\bar{x}^k) - \Psi^*]$. We show that these bounds are not improvable when the dimension n is large enough by developing new lower complexity bounds for randomized incremental gradient methods. Extensions of the randomized primal-dual gradient method to non-strongly convex, nonsmooth, and unbounded problems are also discussed in this chapter. It should be noted that in this paper we focus on the theoretic convergence properties of these primal-dual gradient methods, and the algorithmic parameters were chosen in a conservative manner and were dependent on a few problem parameters, e.g., L and μ. In the future, it will be interesting to develop more adaptive versions of these algorithms which do not require the explicit estimation about L and μ.

CHAPTER 3
RANDOM GRADIENT EXTRAPOLATION FOR DISTRIBUTED AND STOCHASTIC OPTIMIZATION

3.1 Overview

In this chapter, we consider a class of finite-sum convex optimization problems defined over a distributed multiagent network with m agents connected to a central server. In particular, We set the simple convex function $h(\cdot) = 0$ in (1.1.1), and hence the objective function reduced to

$$\psi^* := \min_{x \in X} \left\{ \psi(x) := \tfrac{1}{m} \textstyle\sum_{i=1}^m f_i(x) + \mu w(x) \right\}. \tag{3.1.1}$$

We also relax the smoothness assumption enforced on f_i in Chapter 2, i.e., we assume that $f_i : X \to \mathbb{R}, \;\; i = 1, \ldots, m$, are smooth convex functions with Lipschitz continuous gradients over X, i.e., $\exists L_i \geq 0$ such that

$$\|\nabla f_i(x_1) - \nabla f_i(x_2)\|_* \leq L_i \|x_1 - x_2\|, \;\; \forall x_1, x_2 \in X. \tag{3.1.2}$$

Our major contribution is to develop a new randomized incremental gradient algorithm, namely random gradient extrapolation method (RGEM), which does not require any exact gradient evaluation even for the initial point, but can achieve the optimal $\mathcal{O}(\log(1/\epsilon))$ complexity bound in terms of the total number of gradient evaluations of component functions to solve the finite-sum problems. Furthermore, we demonstrate that for stochastic finite-sum optimization problems, RGEM maintains the optimal $\mathcal{O}(1/\epsilon)$ complexity (up to a certain logarithmic factor) in terms of the number of stochastic gradient computations, but attains an $\mathcal{O}(\log(1/\epsilon))$ complexity in terms of communication rounds (each round involves only one agent). It is worth noting that the former bound is independent of the number of

agents m, while the latter one only linearly depends on m or even $\sqrt{m}$ for ill-conditioned problems. To the best of our knowledge, this is the first time that these complexity bounds have been obtained for distributed and stochastic optimization problems. Moreover, our algorithms were developed based on a novel dual perspective of Nesterov's accelerated gradient method.

The rest of this chapter is organized as follows. In Section 3.2 we present the proposed random gradient extrapolation methods (RGEM), and their convergence properties for solving (3.1.1) and (1.1.5). In order to provide more insights into the design of the algorithmic scheme of RGEM, we provide an introduction to the gradient extrapolation method (GEM) and its relation to the primal-dual gradient method, as well as Nesterov's method in Section 3.3. Section 3.4 is devoted to the convergence analysis of RGEM. Some concluding remarks are made in Section 5.

3.1.1 Notation and Terminology

We use $\|\cdot\|$ to denote a general norm in $\mathbb{R}^n$ without specific mention. We also use $\|\cdot\|_*$ to denote the conjugate norm of $\|\cdot\|$. For any $p \geq 1$, $\|\cdot\|_p$ denotes the standard p-norm in $\mathbb{R}^n$, i.e., $\|x\|_p^p = \sum_{i=1}^n |x_i|^p$, for any $x \in \mathbb{R}^n$. For any convex function h, $\partial h(x)$ is the set of subdifferential at x. For a given strongly convex function w with modulus 1 (see (3.1.1)), we define a *prox-function* associated with w as

$$P(x^0, x) \equiv P_w(x^0, x) := w(x) - \left[w(x^0) + \langle w'(x^0), x - x^0\rangle\right], \tag{3.1.3}$$

where $w'(x^0) \in \partial w(x^0)$ is an arbitrary subgradient of w at x^0. By the strong convexity of w, we have

$$P(x^0, x) \geq \frac{1}{2}\|x - x^0\|^2, \ \ \forall x, x^0 \in X. \tag{3.1.4}$$

It should be pointed out that the prox-function $P(\cdot,\cdot)$ described above is a generalized Bregman distance in the sense that w is not necessarily differentiable. This is different

from the standard definition for Bregman distance [86, 87, 88, 89, 100]. Throughout this chapter, we assume that the prox-mapping associated with X and w, given by

$$\mathcal{M}_X(g, x^0, \eta) := \operatorname{argmin}_{x \in X} \left\{ \langle g, x \rangle + \mu w(x) + \eta P(x^0, x) \right\}, \tag{3.1.5}$$

is easily computable for any $x^0 \in X, g \in \mathbb{R}^n, \mu \geq 0, \eta > 0$. For any real number r, $\lceil r \rceil$ and $\lfloor r \rfloor$ denote the nearest integer to r from above and below, respectively. $\mathbb{R}_+$ and $\mathbb{R}_{++}$, respectively, denote the set of nonnegative and positive real numbers.

3.2 Algorithms and Main Results

This section contains three subsections. We first present in Subsection 3.2.1 an optimal random gradient extrapolation method (RGEM) for solving the distributed finite-sum problem in (3.1.1), and then discuss in Subsection 3.2.2, a stochastic version of RGEM for solving the stochastic finite-sum problem in (1.1.5). Subsection 3.2.3 is devoted to the implementation of RGEM in a distributed setting and the discussion about its communication complexity.

3.2.1 RGEM for Deterministic Finite-sum Optimization

The basic scheme of RGEM is formally stated in Algorithm 5. This algorithm simply initializes the gradient as $y_i^{-1} = y_i^0 = \mathbf{0},\ i = 1, \ldots, m$. At each iteration, RGEM requires the new gradient information of only one randomly selected component function f_i, but maintains m pairs of search points and gradients $(\underline{x}_i^t, y_i^t)$, $i = 1, \ldots, m$, which are stored by their corresponding agents in the distributed network. More specifically, it first performs a gradient extrapolation step in (3.2.6) and the primal proximal mapping in (3.2.7). Then a randomly selected block $\underline{x}_{i_t}^t$ is updated in (3.2.8) and the corresponding component gradient ∇f_{i_t} is computed in (3.2.9). As can be seen from Algorithm 5, RGEM does not require any exact gradient evaluations.

Algorithm 5 A random gradient extrapolation method (RGEM)

Input: Let $x^0 \in X$, and the nonnegative parameters $\{\alpha_t\}$, $\{\eta_t\}$, and $\{\tau_t\}$ be given.

Initialization:

Set $\underline{x}_i^0 = x^0,\ y_i^{-1} = y_i^0 = \mathbf{0},\ i = 1, \ldots, m.$ ▷ No exact gradient evaluation for initialization

for $t = 1, \ldots, k$ **do**

Choose i_t according to Prob$\{i_t = i\} = \frac{1}{m}, i = 1, \ldots, m.$

$$\tilde{y}_i^t = y_i^{t-1} + \alpha_t(y_i^{t-1} - y_i^{t-2}), \forall i, \tag{3.2.6}$$

$$x^t = \mathcal{M}_X(\tfrac{1}{m}\textstyle\sum_{i=1}^m \tilde{y}_i^t, x^{t-1}, \eta_t), \tag{3.2.7}$$

$$\underline{x}_i^t = \begin{cases} (1+\tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1}), & i = i_t, \\ \underline{x}_i^{t-1}, & i \neq i_t. \end{cases} \tag{3.2.8}$$

$$y_i^t = \begin{cases} \nabla f_i(\underline{x}_i^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{3.2.9}$$

end for

Output: For some $\theta_t > 0$, $t = 1, \ldots, k$, set

$$\underline{x}^k := (\textstyle\sum_{t=1}^k \theta_t)^{-1} \sum_{t=1}^k \theta_t x^t. \tag{3.2.10}$$

Note that the computation of x^t in (3.2.7) requires an involved computation of $\frac{1}{m}\sum_{i=1}^m \tilde{y}_i^t$. In order to save computational time when implementing this algorithm, we suggest to compute this quantity in a recursive manner as follows. Let us denote $g^t \equiv \frac{1}{m}\sum_{i=1}^m y_i^t,\ t = 1, \ldots, k$. Clearly, in view of the fact that $y_i^t = y_i^{t-1}, \forall i \neq i_t$, we have

$$g^t = g^{t-1} + \tfrac{1}{m}(y_{i_t}^t - y_{i_t}^{t-1}). \tag{3.2.11}$$

Also, by the definition of g^t and (3.2.6), we have

$$\tfrac{1}{m}\textstyle\sum_{i=1}^m \tilde{y}_i^t = \tfrac{1}{m}\sum_{i=1}^m y_i^{t-1} + \tfrac{\alpha_t}{m}(y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}) = g^{t-1} + \tfrac{\alpha_t}{m}(y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}). \tag{3.2.12}$$

Using these two ideas mentioned above, we can compute $\frac{1}{m}\sum_{i=1}^m \tilde{y}_i^t$ in two steps: i) initialize $g^0 = \mathbf{0}$, and update g^t as in (3.2.11) after the gradient evaluation step (3.2.9); ii)

replace (3.2.6) by (3.2.12) to compute $\frac{1}{m}\sum_{i=1}^{m}\tilde{y}_i^t$. Also note that the difference $y_{i_t}^t - y_{i_t}^{t-1}$ can be saved as it is used in both (3.2.11) and (3.2.12) for the next iteration. These enhancements will be incorporated into the distributed setting in Subsection 3.2.3 to possibly save communication costs.

It is also interesting to observe the differences between RGEM and RPDG discussed in Chapter 2. RGEM has only one extrapolation step (3.2.6) which combines two types of predictions. One is to predict future gradients using historic data, and the other is to obtain an estimator of the current exact gradient of f from the randomly updated gradient information of f_i. However, RPDG method needs two extrapolation steps in both the primal and dual spaces. Due to the existence of the primal extrapolation step, RPDG cannot guarantee the search points where it performs gradient evaluations to fall within the feasible set X. Hence, it requires the assumption that f_i's are differentiable with Lipschitz continuous gradients over $\mathbb{R}^n$. Such a strong assumption is not required by RGEM, since all the primal iterates generated by RGEM stay within the feasible region X. As a result, RGEM can deal with a much wider class of problems than RPDG. Moreover, in RGEM we do not need to compute the exact gradients at the initialization point $\underline{x}_i^0$, but simply set them as $y_i^0 = 0$. It can be seen that under the L-smooth assumption on gradients (cf. (1.1.4)), there exists $0 \leq \sigma_0 < +\infty$ such that

$$\frac{1}{m}\sum_{i=1}^{m}\|\nabla f_i(x^0)\|_*^2 = \sigma_0^2. \tag{3.2.13}$$

We now provide a constant step-size policy for RGEM to solve strongly convex problems given in the form of (3.1.1) and show that the resulting algorithm exhibits an optimal linear rate of convergence in Theorem 3.2.1. The proof of Theorem 3.2.1 can be found in Subsection 3.4.1.

Theorem 3.2.1 *Let x^* be an optimal solution of (3.1.1), x^k and $\underline{x}^k$ be defined in* (3.2.7)

and (3.2.10), respectively, and $\hat{L} = \max_{i=1,\ldots,m} L_i$. *Also let* $\{\tau_t\}$, $\{\eta_t\}$ *and* $\{\alpha_t\}$ *be set to*

$$\tau_t \equiv \tau = \tfrac{1}{m(1-\alpha)} - 1, \quad \eta_t \equiv \eta = \tfrac{\alpha}{1-\alpha}\mu, \quad \textit{and} \quad \alpha_t \equiv m\alpha. \tag{3.2.14}$$

If (3.2.13) *holds and* α *is set as*

$$\alpha = 1 - \tfrac{1}{m+\sqrt{m^2+16m\hat{L}/\mu}}, \tag{3.2.15}$$

then

$$\mathbb{E}[P(x^k, x^*)] \leq \tfrac{2\Delta_{0,\sigma_0}\alpha^k}{\mu}, \tag{3.2.16}$$

$$\mathbb{E}[\psi(\underline{x}^k) - \psi(x^*)] \leq 16\max\left\{m, \tfrac{\hat{L}}{\mu}\right\}\Delta_{0,\sigma_0}\alpha^{k/2}, \tag{3.2.17}$$

where

$$\Delta_{0,\sigma_0} := \mu P(x^0, x^*) + \psi(x^0) - \psi^* + \tfrac{\sigma_0^2}{m\mu}. \tag{3.2.18}$$

In view of Theorem 3.2.1, we can provide bounds on the total number of gradient evaluations performed by RGEM to find a stochastic ϵ-solution of problem (3.1.1), i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[\psi(\bar{x}) - \psi^*] \leq \epsilon$. Theorem 3.2.1 implies the number of gradient evaluations of f_i performed by RGEM to find a stochastic ϵ-solution of (3.1.1) can be bounded by

$$K(\epsilon, C, \sigma_0^2) = 2\left(m + \sqrt{m^2+16mC}\right)\log\tfrac{16\max\{m,C\}\Delta_{0,\sigma_0}}{\epsilon} = \mathcal{O}\left\{\left(m + \sqrt{\tfrac{m\hat{L}}{\mu}}\right)\log\tfrac{1}{\epsilon}\right\}. \tag{3.2.19}$$

Here $C = \hat{L}/\mu$. Therefore, whenever $\sqrt{mC}\log(1/\epsilon)$ is dominating, and L_f and $\hat{L}$ are in the same order of magnitude, RGEM can save up to $\mathcal{O}(\sqrt{m})$ gradient evaluations of the component function f_i than the optimal deterministic first-order methods. More specifically, RGEM does not require any exact gradient computation and its communication cost is similar to pure stochastic gradient descent. To the best of our knowledge, it is the first

time that such an optimal RIG method is presented for solving (3.1.1) in the literature. It should be pointed out that while the rates of convergence of RGEM obtained in Theorem 3.2.1 is stated in terms of expectation, we can develop large-deviation results for these rates of convergence using similar techniques in Chapter 2 for solving strongly convex problems.

Furthermore, if a one-time exact gradient evaluation is available at the initial point, i.e., $y_i^{-1} = y_i^0 = \nabla f_i(x^0),\ i = 1, \ldots, m$, we can employ a more aggressive stepsize policy with

$$\alpha = 1 - \frac{2}{m + \sqrt{m^2 + 8m\hat{L}/\mu}}.$$

Similarly, we can demonstrate that the number of gradient evaluations of f_i performed by RGEM with this initialization method to find a stochastic ϵ-solution can be bounded by

$$\left(m + \sqrt{m^2 + 8mC}\right) \log\left(\frac{6\max\{m, C\}\Delta_{0,0}}{\epsilon}\right) + m = \mathcal{O}\left\{\left(m + \sqrt{\frac{m\hat{L}}{\mu}}\right)\log\frac{1}{\epsilon}\right\}.$$

It is worth noting that according to the parameter setting in (3.2.14), we have

$$\eta = (\tfrac{1}{1-\alpha} - 1)\mu = \left(m + \sqrt{m^2 + 16m\hat{L}/\mu}\right)\mu - \mu = \Omega(m\mu + \sqrt{mL\mu}).$$

In some statistical learning applications with $L2$ regularization (i.e., $\omega(x) = \|x\|_2^2/2$), one usually chooses $\mu = \Omega(1/m)$. For these applications, the stepsize of RGEM is in the order of $1/\sqrt{L}$, while SAGA and SVRG use stepsizes in the order of $1/L$. Hence the stepsize of RGEM can be larger than those of SAGA and SVRG whenever $L \geq 1$.

3.2.2 RGEM for Stochastic Finite-sum Optimization

We discuss in this subsection the stochastic finite-sum optimization and online learning problems, where only noisy gradient information of f_i can be accessed via a stochastic first-order ($\mathcal{SFO}$) oracle. In particular, for any given point $\underline{x}_i^t \in X$, the $\mathcal{SFO}$ oracle

outputs a vector $G_i(\underline{x}_i^t, \xi_i^t)$ s.t.

$$\mathbb{E}_\xi[G_i(\underline{x}_i^t, \xi_i^t)] = \nabla f_i(\underline{x}_i^t),\ i = 1, \ldots, m, \tag{3.2.20}$$

$$\mathbb{E}_\xi[\|G_i(\underline{x}_i^t, \xi_i^t) - \nabla f_i(\underline{x}_i^t)\|_*^2] \leq \sigma^2,\ i = 1, \ldots, m. \tag{3.2.21}$$

We also assume that throughout this subsection that the $\|\cdot\|$ is associated with the inner product $\langle \cdot, \cdot \rangle$.

As shown in Algorithm 6, the RGEM for stochastic finite-sum optimization is naturally obtained by replacing the gradient evaluation of f_i in Algorithm 5 (see (3.2.9)) with a stochastic gradient estimator of f_i given in (3.2.22). In particular, at each iteration, we collect B_t number of stochastic gradients of only one randomly selected component f_i and take their average as the stochastic estimator of ∇f_i. Moreover, it needs to be mentioned that the way RGEM initializes gradients, i.e, $y^{-1} = y^0 = \mathbf{0}$, is very important for stochastic optimization, since it is usually impossible to compute exact gradient for expectation functions even at the initial point.

Algorithm 6 RGEM for stochastic finite-sum optimization

This algorithm is the same as Algorithm 5 except that (3.2.9) is replaced by

$$y_i^t = \begin{cases} \frac{1}{B_t}\sum_{j=1}^{B_t} G_i(\underline{x}_i^t, \xi_{i,j}^t), & i = i_t, \\ y_i^{t-1}, & i \neq i_t. \end{cases} \tag{3.2.22}$$

Here, $G_i(\underline{x}_i^t, \xi_{i,j}^t)$, $j = 1, \ldots, B_t$, are stochastic gradients of f_i computed by the $\mathcal{SFO}$ oracle at $\underline{x}_i^t$.

Under the standard assumptions in (3.2.20) and (3.2.21) for stochastic optimization, and with proper choices of algorithmic parameters, Theorem 3.2.2 shows that RGEM can achieve the optimal $\mathcal{O}\{\sigma^2/\mu^2\epsilon\}$ rate of convergence (up to a certain logarithmic factor) for solving strongly convex problems given in the form of (1.1.5) in terms of the number of

stochastic gradients of f_i. The proof of the this result can be found in Subsection 3.4.2.

Theorem 3.2.2 *Let x^* be an optimal solution of (1.1.5), x^k and $\underline{x}^k$ be generated by Algorithm 6, and $\hat{L} = \max_{i=1,\ldots,m} L_i$. Suppose that σ_0 and σ are defined in* (3.2.13) *and* (3.2.21)*, respectively. Given the iteration limit k, let $\{\tau_t\}$, $\{\eta_t\}$ and $\{\alpha_t\}$ be set to* (3.2.14) *with α being set as* (3.2.15)*, and we also set*

$$B_t = \lceil k(1-\alpha)^2\alpha^{-t} \rceil,\ t = 1, \ldots, k, \tag{3.2.23}$$

then

$$\mathbb{E}[P(x^k, x^*)] \leq \tfrac{2\alpha^k \Delta_{0,\sigma_0,\sigma}}{\mu}, \tag{3.2.24}$$

$$\mathbb{E}[\psi(\underline{x}^k) - \psi(x^*)] \leq 16 \max\left\{m, \tfrac{\hat{L}}{\mu}\right\} \Delta_{0,\sigma_0,\sigma} \alpha^{k/2}, \tag{3.2.25}$$

where the expectation is taken w.r.t. $\{i_t\}$ and $\{\xi_i^t\}$ and

$$\Delta_{0,\sigma_0,\sigma} := \mu P(x^0, x^*) + \psi(x^0) - \psi(x^*) + \tfrac{\sigma_0^2/m + 5\sigma^2}{\mu}. \tag{3.2.26}$$

In view of (3.2.25), the number of iterations performed by RGEM to find a stochastic ϵ-solution of (1.1.5), can be bounded by

$$\hat{K}(\epsilon, C, \sigma_0^2, \sigma^2) := 2\left(m + \sqrt{m^2 + 16mC}\right) \log \tfrac{16 \max\{m, C\} \Delta_{0,\sigma_0,\sigma}}{\epsilon}. \tag{3.2.27}$$

Furthermore, in view of (3.2.24) this iteration complexity bound can be improved to

$$\bar{K}(\epsilon, \alpha, \sigma_0^2, \sigma^2) := \log_{1/\alpha} \tfrac{2\tilde{\Delta}_{0,\sigma_0,\sigma}}{\mu\epsilon}, \tag{3.2.28}$$

in terms of finding a point $\bar{x} \in X$ s.t. $\mathbb{E}[P(\bar{x}, x^*)] \leq \epsilon$. Therefore, the corresponding number of stochastic gradient evaluations performed by RGEM for solving problem (1.1.5)

can be bounded by

$$\sum_{t=1}^{k} B_t \leq k\sum_{t=1}^{k}(1-\alpha)^2\alpha^{-t} + k = \mathcal{O}\left\{\left(\frac{\Delta_{0,\sigma_0,\sigma}}{\mu\epsilon} + m + \sqrt{mC}\right)\log\frac{\Delta_{0,\sigma_0,\sigma}}{\mu\epsilon}\right\}, \quad (3.2.29)$$

which together with (3.2.26) imply that the total number of required stochastic gradients or samples of the random variables $\xi_i,\ i = 1,\ldots,m$, can be bounded by

$$\tilde{\mathcal{O}}\left\{\frac{\sigma_0^2/m+\sigma^2}{\mu^2\epsilon} + \frac{\mu P(x^0,x^*)+\psi(x^0)-\psi^*}{\mu\epsilon} + m + \sqrt{\frac{m\hat{L}}{\mu}}\right\}.$$

Observe that this bound does not depend on the number of terms m for small enough ϵ. To the best of our knowledge, it is the first time that such a convergence result is established for RIG algorithms to solve distributed stochastic finite-sum problems. This complexity bound in fact is in the same order of magnitude (up to a logarithmic factor) as the complexity bound achieved by the optimal accelerated stochastic approximation methods [11, 12, 10], which uniformly sample all the random variables $\xi_i,\ i = 1,\ldots,m$. However, this latter approach will thus involve much higher communication costs in the distributed setting (see Subsection 3.2.3 for more discussions).

3.2.3 RGEM for Distributed Optimization and Machine Learning

This subsection is devoted to RGEMs (see Algorithm 5 and Algorithm 6) from two different perspectives, i.e., the server and the activated agent under a distributed setting. We also discuss the communication costs incurred by RGEM under this setting.

Both the server and agents in the distributed network start with the same global initial point x^0, i.e., $\underline{x}_i^0 = x^0,\ i = 1,\ldots,m$, and the server also sets $\Delta y = \mathbf{0}$ and $g^0 = \mathbf{0}$. During the process of RGEM, the server updates iterate x^t and calculates the output solution $\underline{x}^k$ (cf. (3.2.10)) which is given by $\mathrm{sum}x/\mathrm{sum}\theta$. Each agent only stores its local variable $\underline{x}_i^t$ and updates it according to the information received from the server (i.e., x^t) when activated. The activated agent also needs to upload the changes of gradient Δy_i to the server. Note

that line 5 of RGEM from the i_t-th agent's perspective is optional if the agent saves historic gradient information from the last update.

RGEM The server's perspective

1: **while** $t \leq k$ **do**
2: $x^t \leftarrow \mathcal{M}_X(g^{t-1} + \frac{\alpha_t}{m}\Delta y, x^{t-1}, \eta_t)$
3: $\mathrm{sum}x \leftarrow \mathrm{sum}x + \theta_t x^t$
4: $\mathrm{sum}\theta \leftarrow \mathrm{sum}\theta + \theta_t$
5: Send signal to the i_t-th agent where i_t is selected uniformly from $\{1, \ldots, m\}$
6: **if** i_t-th agent is responsive **then**
7: Send current iterate x^t to i_t-th agent
8: **if** Receive feedback Δy **then**
9: $g^t \leftarrow g^{t-1} + \Delta y$
10: $t \leftarrow t + 1$
11: **else goto** *Line 5*
12: **end if**
13: **else goto** *Line 5*
14: **end if**
15: **end while**

RGEM The activated i_t-th agent's perspective

1: Download the current iterate x^t from the server
2: **if** $t = 1$ **then**
3: $y_i^{t-1} \leftarrow \mathbf{0}$
4: **else**
5: $y_i^{t-1} \leftarrow \nabla f_i(\underline{x}_i^{t-1})$ ▷ Optional
6: **end if**
7: $\underline{x}_i^t \leftarrow (1 + \tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1})$
8: $y_i^t \leftarrow \nabla f_i(\underline{x}_i^t)$
9: Upload the local changes to the server, i.e., $\Delta y_i = y_i^t - y_i^{t-1}$

We now add some remarks about the potential benefits of RGEM for distributed optimization and machine learning. Firstly, since RGEM does not require any exact gradient evaluation of f, it does not need to wait for the responses from all agents in order to compute an exact gradient. Each iteration of RGEM only involves communication between the server and the activated i_t-th agent. In fact, RGEM will move to the next iteration in case no response is received from the i_t-th agent. This scheme works under the assumption that the

probability for any agent being responsive or available at a certain point of time is equal. However, all other optimal RIG algorithms, except RPDG discussed in Chapter 2, need the exact gradient information from all network agents once in a while, which incurs high communication costs and synchronous delays as long as one agent is not responsive. Even RPDG requires a full round of communications and synchronization at the initial point.

Secondly, since each iteration of RGEM involves only constant number of communication rounds between the server and one selected agent, the communication complexity for RGEM under distributed setting can be bounded by

$$\mathcal{O}\left\{\left(m+\sqrt{\tfrac{m\hat{L}}{\mu}}\right)\log\tfrac{1}{\epsilon}\right\}.$$

Therefore, it can save up to $\mathcal{O}\{\sqrt{m}\}$ rounds of communication than the optimal deterministic first-order methods.

For solving distributed stochastic finite-sum optimization problems (1.1.5), RGEM from the i_t-th agent's perspective will be slightly modified as follows.

RGEM The activated i_t-th agent's perspective for solving (1.1.5)

1: Download the current iterate x^t from the server

2: **if** $t = 1$ **then**

3: $\quad y_i^{t-1} \leftarrow \mathbf{0}$ ▷ Assuming RGEM saves y_i^{t-1} for $t \geq 2$ at the latest update

4: **end if**

5: $\underline{x}_i^t \leftarrow (1+\tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1})$

6: $y_i^t \leftarrow \frac{1}{B_t}\sum_{j=1}^{B_t} G_i(\underline{x}_i^t, \xi_{i,j}^t)$ ▷ B_t is the batch size, and G_i's are the stochastic gradients given by $\mathcal{SFO}$

7: Upload the local changes to the server, i.e., $\Delta y_i = y_i^t - y_i^{t-1}$

Similar to the case for the deterministic finite-sum optimization, the total number of

communication rounds performed by the above RGEM can be bounded by

$$\mathcal{O}\left\{\left(m+\sqrt{\frac{m\hat{L}}{\mu}}\right)\log\frac{1}{\epsilon}\right\},$$

for solving (1.1.5). Each round of communication only involves the server and a randomly selected agent. This communication complexity seems to be optimal, since it matches the lower complexity bound (1.1.12) established in Section 2.3.3 of Chapter 2. Moreover, the sampling complexity, i.e., the total number of samples to be collected by all the agents, is also nearly optimal and comparable to the case when all these samples are collected in a centralized location and processed by an optimal stochastic approximation method. On the other hand, if one applies an existing optimal stochastic approximation method to solve the distributed stochastic optimization problem, the communication complexity will be as high as $\mathcal{O}(1/\sqrt{\epsilon})$, which is much worse than RGEM.

3.3 Gradient Extrapolation Method: Dual of Nesterov's Acceleration

Our goal in this section is to introduce a new algorithmic framework, referred to as the gradient extrapolation method (GEM), for solving the convex optimization problem given by

$$\psi^* := \min_{x\in X}\{\psi(x) := f(x)+\mu w(x)\}. \tag{3.3.1}$$

We show that GEM can be viewed as a dual of Nesterov's accelerated gradient method although these two algorithms appear to be quite different. Moreover, GEM possess some nice properties which enable us to develop and analyze the random gradient extrapolation method for distributed and stochastic optimization.

3.3.1 Generalized Bregman Distance

In this subsection, we provide a brief introduction to the generalized Bregman distance defined in (3.1.3). Note that whenever w is non-differentiable, we need to specify a particular selection of the subgradient w' before performing the prox-mapping. We assume throughout this chapter that such a selection of w' is defined recursively as follows. Denote $x^1 \equiv \mathcal{M}_X(g, x^0, \eta)$. By the optimality condition of (3.1.5), we have

$$g + (\mu + \eta)w'(x^1) - \eta w'(x^0) \in \mathcal{N}_X(x^1),$$

where $\mathcal{N}_X(x^1) := \{v \in \mathbb{R}^n : v^T(x - x^1) \leq 0, \forall x \in X\}$ denotes the normal cone of X at x^1. Once such a $w'(x^1)$ satisfying the above relation is identified, we will use it as a subgradient when defining $P(x^1, x)$ in the next iteration. Note that such a subgradient can be identified as long as x^1 is obtained, since it satisfies the optimality condition of (3.1.5).

3.3.2 The Algorithm

As shown in Algorithm 7, GEM starts with a gradient extrapolation step (3.3.2) to compute $\tilde{g}^t$ from the two previous gradients g^{t-1} and g^{t-2}. Based on $\tilde{g}^t$, it performs a proximal gradient descent step in (3.3.3) and updates the output solution $\underline{x}^t$. Finally, the gradient at $\underline{x}^t$ is computed for gradient extrapolation in the next iteration. This algorithm is a special case of RGEM in Algorithm 5 (with $m = 1$).

Algorithm 7 An optimal gradient extrapolation method (GEM)

Input: Let $x^0 \in X$, and the nonnegative parameters $\{\alpha_t\}$, $\{\eta_t\}$, and $\{\tau_t\}$ be given.

Set $\underline{x}^0 = x^0$ and $g^{-1} = g^0 = \nabla f(x^0)$.

for $t = 1, 2, \ldots, k$ **do**

$$\tilde{g}^t = \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1}. \tag{3.3.2}$$

$$x^t = \mathcal{M}_X(\tilde{g}^t, x^{t-1}, \eta_t). \tag{3.3.3}$$

$$\underline{x}^t = \left(x^t + \tau_t \underline{x}^{t-1}\right)/(1+\tau_t). \tag{3.3.4}$$

$$g^t = \nabla f(\underline{x}^t). \tag{3.3.5}$$

end for

Output: $\underline{x}^k$.

We now show that GEM can be viewed as the dual of the well-known Nesterov's accelerated gradient (NAG) method as studied in Chapter 2. To see such a relationship, we will first rewrite GEM in a primal-dual form. Let us consider the dual space $\mathcal{G}$, where the gradients of f reside, and equip it with the conjugate norm $\|\cdot\|_*$. Let $J_f : \mathcal{G} \to \mathbb{R}$ be the conjugate function of f such that $f(x) := \max_{g\in\mathcal{G}}\{\langle x, g\rangle - J_f(g)\}$. We can reformulate the original problem in (3.3.1) as the following saddle point problem:

$$\psi^* := \min_{x\in X}\left\{\max_{g\in\mathcal{G}}\{\langle x, g\rangle - J_f(g)\} + \mu\, w(x)\right\}. \tag{3.3.6}$$

It is clear that J_f is strongly convex with modulus $1/L_f$ w.r.t. $\|\cdot\|_*$ (See Chapter E in [90] for details). Therefore, we can define its associated dual generalized Bregman distance and dual prox-mappings as in (2.2.5) and (2.2.6). By Lemma 2.2.1, we can see that the GEM iteration can be written in a primal-dual form. Given $(x^0, g^{-1}, g^0) \in X \times \mathcal{G} \times \mathcal{G}$, it updates

(x^t, g^t) by

$$\tilde{g}^t = \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1}, \tag{3.3.7}$$

$$x^t = \mathcal{M}_X(\tilde{g}^t, x^{t-1}, \eta_t), \tag{3.3.8}$$

$$g^t = \mathcal{M}_{\mathcal{G}}(-x^t, g^{t-1}, \tau_t), \tag{3.3.9}$$

with a specific selection of $J_f'(g^{t-1}) = \underline{x}^{t-1}$ in $D_f(g^{t-1}, g)$. Indeed, by denoting $\underline{x}^0 = x^0$, we can easily see from $g^0 = \nabla f(\underline{x}^0)$ that $\underline{x}^0 \in \partial J_f(g^0)$. Now assume that $g^{t-1} = \nabla f(\underline{x}^{t-1})$ and hence that $\underline{x}^{t-1} \in \partial J_f(g^{t-1})$. By the definition of g^t in (3.3.9) and Lemma 2.2.1, we conclude that $g^t = \nabla f(\underline{x}^t)$ with $\underline{x}^t = (x^t + \tau_t \underline{x}^{t-1})/(1 + \tau_t)$, which are exactly the definitions given in (3.3.4) and (3.3.5).

Recall that in a simple version of the NAG method (e.g., [6, 9, 10, 11, 12, 92]), given $(x^{t-1}, \bar{x}^{t-1}) \in X \times X$, it updates $(x^t, \bar{x}^t)$ by (2.2.15)-(2.2.18). Moreover, we have shown in Chapter 2 that (2.2.15)-(2.2.18) can be viewed as a specific instantiation of the primal-dual updates (2.2.8)-(2.2.10). Comparing (3.3.7)-(3.3.9) with (2.2.8)-(2.2.10), we can clearly see that GEM is a dual version of NAG, obtained by switching the primal and dual variables in each equation of (2.2.8)-(2.2.10). The major difference exists in that the extrapolation step in GEM is performed in the dual space while the one in NAG is performed in the primal space. In fact, extrapolation in the dual space will help us to greatly simplify and further enhance the randomized incremental gradient methods developed in Chapter 2 based on NAG. Another interesting fact is that in GEM, the gradients are computed for the output solutions $\{\underline{x}^t\}$. On the other hand, the output solutions in the NAG method are given by $\{\bar{x}^t\}$ while the gradients are computed for the extrapolation sequence $\{\underline{x}^t\}$.

3.3.3 Convergence of GEM

Our goal in this subsection is to establish the convergence properties of the GEM method for solving (3.3.1). Observe that our analysis is carried out completely in the primal space

and does not rely on the primal-dual interpretation described in the previous section. This type of analysis technique appears to be new for solving problem (3.3.1) in the literature as it also differs significantly from that of NAG.

We first establish some general convergence properties for GEM for both smooth convex ($\mu = 0$) and strongly convex cases ($\mu > 0$).

Theorem 3.3.3 *Suppose that* $\{\eta_t\}$, $\{\tau_t\}$, *and* $\{\alpha_t\}$ *in GEM satisfy*

$$\theta_{t-1} = \alpha_t \theta_t, \;\; t = 2, \ldots, k, \tag{3.3.10}$$

$$\theta_t \eta_t \leq \theta_{t-1}(\mu + \eta_{t-1}), \;\; t = 2, \ldots, k, \tag{3.3.11}$$

$$\theta_t \tau_t = \theta_{t-1}(1 + \tau_{t-1}), \; t = 2, \ldots, k, \tag{3.3.12}$$

$$\alpha_t L_f \leq \tau_{t-1} \eta_t, \;\; t = 2, \ldots, k, \tag{3.3.13}$$

$$2L_f \leq \tau_k(\mu + \eta_k), \tag{3.3.14}$$

for some $\theta_t \geq 0$, $t = 1, \ldots, k$. *Then, for any* $k \geq 1$ *and any given* $x \in X$, *we have*

$$\theta_k(1 + \tau_k)[\psi(\underline{x}^k) - \psi(x)] + \tfrac{\theta_k(\mu+\eta_k)}{2} P(x^k, x) \leq \theta_1 \tau_1 [\psi(x^0) - \psi(x)] + \theta_1 \eta_1 P(x^0, x). \tag{3.3.15}$$

Proof. Applying Lemma 2.5.16 to (3.3.3), we obtain

$$\langle x^t - x, \alpha_t(g^{t-1} - g^{t-2}) + g^{t-1} \rangle + \mu w(x^t) - \mu w(x) \leq \eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t). \tag{3.3.16}$$

Moreover, using the definition of ψ, the convexity of f, and the fact that $g^t = \nabla f(\underline{x}^t)$, we have

$$\begin{aligned}(1 + \tau_t) f(\underline{x}^t) + \mu w(x^t) - \psi(x) &\leq (1 + \tau_t) f(\underline{x}^t) + \mu w(x^t) - \mu w(x) - [f(\underline{x}^t) + \langle g^t, x - \underline{x}^t \rangle] \\ &= \tau_t [f(\underline{x}^t) - \langle g^t, \underline{x}^t - \underline{x}^{t-1} \rangle] - \langle g^t, x - x^t \rangle + \mu w(x^t) - \mu w(x)\end{aligned}$$

$$\leq -\tfrac{\tau_t}{2L_f}\|g^t - g^{t-1}\|_*^2 + \tau_t f(\underline{x}^{t-1}) - \langle g^t, x - x^t\rangle + \mu w(x^t) - \mu w(x),$$

where the first equality follows from the definition of $\underline{x}^t$ in (3.3.4), and the last inequality follows from the smoothness of f (see Theorem 2.1.5 in [6]). In view of (3.3.16), we have

$$\begin{aligned}(1+\tau_t) f(\underline{x}^t) + \mu w(x^t) - \psi(x) \leq & -\tfrac{\tau_t}{2L_f}\|g^t - g^{t-1}\|_*^2 + \tau_t f(\underline{x}^{t-1}) \\ & + \langle x^t - x, g^t - g^{t-1} - \alpha_t(g^{t-1} - g^{t-2})\rangle \\ & + \eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t).\end{aligned}$$

Multiplying both sides of the above inequality by θ_t, and summing up the resulting inequalities from $t = 1$ to k, we obtain

$$\begin{aligned}&\textstyle\sum_{t=1}^k \theta_t(1+\tau_t) f(\underline{x}^t) + \sum_{t=1}^k \theta_t[\mu w(x^t) - \psi(x)] \leq -\sum_{t=1}^k \frac{\theta_t \tau_t}{2L_f}\|g^t - g^{t-1}\|_*^2 + \sum_{t=1}^k \theta_t \tau_t f(\underline{x}^{t-1}) \\ &\quad + \textstyle\sum_{t=1}^k \theta_t \langle x^t - x, g^t - g^{t-1} - \alpha_t(g^{t-1} - g^{t-2})\rangle \\ &\quad + \textstyle\sum_{t=1}^k \theta_t[\eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t)]. \qquad (3.3.17)\end{aligned}$$

Now by (3.3.10) and the fact that $g^{-1} = g^0$, we have

$$\begin{aligned}&\textstyle\sum_{t=1}^k \theta_t \langle x^t - x, g^t - g^{t-1} - \alpha_t(g^{t-1} - g^{t-2})\rangle \\ &= \textstyle\sum_{t=1}^k \theta_t[\langle x^t - x, g^t - g^{t-1}\rangle - \alpha_t \langle x^{t-1} - x, g^{t-1} - g^{t-2}\rangle] - \sum_{t=2}^k \theta_t \alpha_t \langle x^t - x^{t-1}, g^{t-1} - g^{t-2}\rangle \\ &= \theta_k \langle x^k - x, g^k - g^{k-1}\rangle - \textstyle\sum_{t=2}^k \theta_t \alpha_t \langle x^t - x^{t-1}, g^{t-1} - g^{t-2}\rangle. \qquad (3.3.18)\end{aligned}$$

Moreover, in view of (3.3.11), (3.3.12) and the definition of $\underline{x}^t$ (3.3.4), we obtain

$$\textstyle\sum_{t=1}^k \theta_t[\eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x)] \overset{(3.3.11)}{\leq} \theta_1 \eta_1 P(x^0, x) - \theta_k(\mu + \eta_k) P(x^k, x), \qquad (3.3.19)$$

$$\textstyle\sum_{t=1}^k \theta_t[(1+\tau_t) f(\underline{x}^t) - \tau_t f(\underline{x}^{t-1})] \overset{(3.3.12)}{=} \theta_k(1+\tau_k) f(\underline{x}^k) - \theta_1 \tau_1 f(\underline{x}^0), \qquad (3.3.20)$$

$$\sum_{t=1}^{k}\theta_t \overset{(3.3.12)}{=} \sum_{t=2}^{k}[\theta_t\tau_t - \theta_{t-1}\tau_{t-1}] + \theta_k = \theta_k(1+\tau_k) - \theta_1\tau_1, \tag{3.3.21}$$

$$\begin{aligned}\theta_k(1+\tau_k)\underline{x}^k &\overset{(3.3.4)}{=} \theta_k(x^k + \tfrac{\tau_k}{1+\tau_{k-1}}x^{k-1} + \cdots + \textstyle\prod_{t=2}^{k}\tfrac{\tau_t}{1+\tau_{t-1}}x^1 + \prod_{t=2}^{k}\tfrac{\tau_t}{1+\tau_{t-1}}\tau_1 x^0)\\ &\overset{(3.3.12)}{=} \textstyle\sum_{t=1}^{k}\theta_t x^t + \theta_1\tau_1 x^0.\end{aligned} \tag{3.3.22}$$

The last two relations, in view of the convexity of $w(\cdot)$, also imply that

$$\theta_k(1+\tau_k)\mu w(\underline{x}^k) \le \textstyle\sum_{t=1}^{k}\theta_t\mu w(x^t) + \theta_1\tau_1\mu w(x^0).$$

Therefore, by (3.3.17) - (3.3.22), and the definition of ψ, we conclude that

$$\begin{aligned}\theta_k(1+\tau_k)[\psi(\underline{x}^k) - \psi(x)] \le \textstyle\sum_{t=2}^{k}\Big[&-\tfrac{\theta_{t-1}\tau_{t-1}}{2L_f}\|g^{t-1} - g^{t-2}\|_*^2\\ &- \theta_t\alpha_t\langle x^t - x^{t-1}, g^{t-1} - g^{t-2}\rangle - \theta_t\eta_t P(x^{t-1}, x^t)\Big]\\ &- \theta_k\left[\tfrac{\tau_k}{2L_f}\|g^k - g^{k-1}\|_*^2 - \langle x^k - x, g^k - g^{k-1}\rangle + (\mu+\eta_k)P(x^k, x)\right]\\ &+ \theta_1\eta_1 P(x^0, x) + \theta_1\tau_1[\psi(x^0) - \psi(x)] - \theta_1\eta_1 P(x^0, x^1).\end{aligned} \tag{3.3.23}$$

By the strong convexity of $P(\cdot,\cdot)$ in (3.1.4), the simple relation that $b\langle u, v\rangle - a\|v\|^2/2 \le b^2\|u\|^2/(2a),\ \forall a > 0$, and the conditions in (3.3.13) and (3.3.14), we have

$$\begin{aligned}&-\textstyle\sum_{t=2}^{k}\left[\tfrac{\theta_{t-1}\tau_{t-1}}{2L_f}\|g^{t-1} - g^{t-2}\|_*^2 + \theta_t\alpha_t\langle x^t - x^{t-1}, g^{t-1} - g^{t-2}\rangle + \theta_t\eta_t P(x^{t-1}, x^t)\right]\\ &\quad\le \textstyle\sum_{t=2}^{k}\tfrac{\theta_t}{2}\left(\tfrac{\alpha_t L_f}{\tau_{t-1}} - \eta_t\right)\|x^{t-1} - x^t\|^2 \le 0\\ &-\theta_k\left[\tfrac{\tau_k}{2L_f}\|g^k - g^{k-1}\|_*^2 - \langle x^k - x, g^k - g^{k-1}\rangle + \tfrac{(\mu+\eta_k)}{2}P(x^k, x)\right]\\ &\quad\le \tfrac{\theta_k}{2}\left(\tfrac{L_f}{\tau_k} - \tfrac{\mu+\eta_k}{2}\right)\|x^k - x\|^2 \le 0.\end{aligned}$$

Using the above relations in (3.3.23), we obtain (3.3.15). ∎

We are now ready to establish the optimal convergence behavior of GEM as a consequence of Theorem 3.3.3. We first provide a constant step-size policy which guarantees an

optimal linear rate of convergence for the strongly convex case ($\mu > 0$).

Corollary 3.3.4 *Let x^* be an optimal solution of* (3.1.1)*, x^k and $\underline{x}^k$ be defined in* (3.3.3) *and* (3.3.4)*, respectively. Suppose that $\mu > 0$, and that $\{\tau_t\}$, $\{\eta_t\}$ and $\{\alpha_t\}$ are set to*

$$\tau_t \equiv \tau = \sqrt{\tfrac{2L_f}{\mu}}, \quad \eta_t \equiv \eta = \sqrt{2L_f\mu}, \quad \textit{and} \quad \alpha_t \equiv \alpha = \tfrac{\sqrt{2L_f/\mu}}{1+\sqrt{2L_f/\mu}}, \ \forall t = 1, \ldots, k. \tag{3.3.24}$$

Then,

$$P(x^k, x^*) \leq 2\alpha^k[P(x^0, x^*) + \tfrac{1}{\mu}(\psi(x^0) - \psi^*)], \tag{3.3.25}$$

$$\psi(\underline{x}^k) - \psi^* \leq \alpha^k \left[\mu P(x^0, x^*) + \psi(x^0) - \psi^*\right]. \tag{3.3.26}$$

Proof. Let us set $\theta_t = \alpha^{-t}$, $t = 1, \ldots, k$. It is easy to check that the selection of $\{\tau_t\}, \{\eta_t\}$ and $\{\alpha_t\}$ in (3.3.24) satisfies conditions (3.3.10)-(3.3.14). In view of Theorem 3.3.3 and (3.3.24), we have

$$\begin{aligned}\psi(\underline{x}^k) - \psi(x^*) + \tfrac{\mu+\eta}{2(1+\tau)}P(x^k, x^*) &\leq \tfrac{\theta_1\tau}{\theta_k(1+\tau)}[\psi(x^0) - \psi(x^*)] + \tfrac{\theta_1\eta}{\theta_k(1+\tau)}P(x^0, x^*) \\ &= \alpha^k[\psi(x^0) - \psi(x^*) + \mu P(x^0, x^*)].\end{aligned}$$

It also follows from the above relation, the fact $\psi(\underline{x}^k) - \psi(x^*) \geq 0$, and (3.3.24) that

$$P(x^k, x^*) \leq \tfrac{2(1+\tau)\alpha^k}{\mu+\eta}[\mu P(x^0, x^*) + \psi(x^0) - \psi(x^*)] = 2\alpha^k[P(x^0, x^*) + \tfrac{1}{\mu}(\psi(x^0) - \psi(x^*))].$$

■

We now provide a stepsize policy which guarantees the optimal rate of convergence for the smooth case ($\mu = 0$). Observe that in smooth case we can estimate the solution quality for the sequence $\{\underline{x}^k\}$ only.

Corollary 3.3.5 *Let x^* be an optimal solution of* (3.1.1)*, and $\underline{x}^k$ be defined in* (3.3.4)*.*

Suppose that $\mu = 0$, and that $\{\tau_t\}$, $\{\eta_t\}$ and $\{\alpha_t\}$ are set to

$$\tau_t = \tfrac{t}{2}, \quad \eta_t = \tfrac{4L_f}{t}, \quad \textit{and} \quad \alpha_t = \tfrac{t}{t+1}, \ \forall t = 1, \ldots, k. \tag{3.3.27}$$

Then,

$$\psi(\underline{x}^k) - \psi(x^*) = f(\underline{x}^k) - f(x^*) \leq \tfrac{2}{(k+1)(k+2)}[f(x^0) - f(x^*) + 8L_f P(x^0, x^*)]. \tag{3.3.28}$$

Proof. Let us set $\theta_t = t + 1$, $t = 1, \ldots, k$. It is easy to check that the parameters in (3.3.27) satisfy conditions (3.3.13)-(3.3.14). In view of (3.3.15) and (3.3.27), we conclude that

$$\psi(\underline{x}^k) - \psi(x^*) \leq \tfrac{2}{(k+1)(k+2)}[\psi(x^0) - \psi(x^*) + 8L_f P(x^0, x^*)].$$

■

In Corollary 3.3.6, we improve the above complexity result in terms of the dependence on $f(x^0) - f(x^*)$ by using a different step-size policy and a slightly more involved analysis for the smooth case ($\mu = 0$).

Corollary 3.3.6 *Let x^* be an optimal solution of* (3.1.1)*, x^k and $\underline{x}^k$ be defined in* (3.3.3) *and* (3.3.4)*, respectively. Suppose that $\mu = 0$, and that $\{\tau_t\}$, $\{\eta_t\}$ and $\{\alpha_t\}$ are set to*

$$\tau_t = \tfrac{t-1}{2}, \quad \eta_t = \tfrac{6L_f}{t}, \quad \textit{and} \quad \alpha_t = \tfrac{t-1}{t}, \ \forall t = 1, \ldots, k. \tag{3.3.29}$$

Then, for any $k \geq 1$,

$$\psi(\underline{x}^k) - \psi(x^*) = f(\underline{x}^k) - f(x^*) \leq \tfrac{12L_f}{k(k+1)} P(x^0, x^*). \tag{3.3.30}$$

Proof. If we set $\theta_t = t, \ t = 1, \ldots, k$. It is easy to check that the parameters in (3.3.29) satisfy conditions (3.3.10)-(3.3.12) and (3.3.14). However, condition (3.3.13) only holds

for $t = 3, \ldots, k$, i.e.,

$$\alpha_t L_f \leq \tau_{t-1}\eta_t, \;\; t = 3, \ldots, k. \tag{3.3.31}$$

In view of (3.3.23) and the fact that $\tau_1 = 0$, we have

$$\begin{aligned}
&\theta_k(1+\tau_k)[\psi(\underline{x}^k) - \psi(x)] \\
&\quad \leq -\theta_2[\alpha_2\langle x^2 - x^1, g^1 - g^0\rangle + \eta_2 P(x^1, x^2)] - \theta_1\eta_1 P(x^0, x^1) \\
&\quad\quad - \textstyle\sum_{t=3}^{k}\left[\frac{\theta_{t-1}\tau_{t-1}}{2L_f}\|g^{t-1} - g^{t-2}\|_*^2 + \theta_t\alpha_t\langle x^t - x^{t-1}, g^{t-1} - g^{t-2}\rangle + \theta_t\eta_t P(x^{t-1}, x^t)\right] \\
&\quad\quad - \theta_k\left[\tfrac{\tau_k}{2L_f}\|g^k - g^{k-1}\|_*^2 - \langle x^k - x, g^k - g^{k-1}\rangle + (\mu + \eta_k)P(x^k, x)\right] + \theta_1\eta_1 P(x^0, x) \\
&\quad \leq \tfrac{\theta_1\alpha_2}{2\eta_2}\|g^1 - g^0\|_*^2 - \tfrac{\theta_1\eta_1}{2}\|x^1 - x^0\|^2 + \textstyle\sum_{t=3}^{k}\frac{\theta_t}{2}\left(\frac{\alpha_t L_f}{\tau_{t-1}} - \eta_t\right)\|x^{t-1} - x^t\|^2 \\
&\quad\quad + \tfrac{\theta_k}{2}\left(\tfrac{L_f}{\tau_k} - \tfrac{\eta_k}{2}\right)\|x^k - x\|^2 + \theta_1\eta_1 P(x^0, x) - \tfrac{\theta_k\eta_k}{2}P(x^k, x) \\
&\quad \leq \tfrac{\theta_1\alpha_2 L_f^2}{2\eta_2}\|\underline{x}^1 - \underline{x}^0\|^2 - \tfrac{\theta_1\eta_1}{2}\|x^1 - x^0\|^2 + \theta_1\eta_1 P(x^0, x) - \tfrac{\theta_k\eta_k}{2}P(x^k, x) \\
&\quad \leq \theta_1\left(\tfrac{\alpha_2 L_f^2}{2\eta_2} - \eta_1\right)\|x^1 - x^0\|^2 + \theta_1\eta_1 P(x^0, x) - \tfrac{\theta_k\eta_k}{2}P(x^k, x),
\end{aligned}$$

where the second inequality follows from the simple relation that $b\langle u, v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a), \; \forall a > 0$ and (3.1.4), the third inequality follows from (3.3.31), (3.3.14), the definition of g^t in (3.3.5) and (1.1.4), and the last inequality follows from the facts that $\underline{x}^0 = x^0$ and $\underline{x}^1 = x^1$ (due to $\tau_1 = 0$). Therefore, by plugging the parameter setting in (3.3.29) into the above inequality, we conclude that

$$\psi(\underline{x}^k) - \psi^* = f(\underline{x}^k) - f(x^*) \leq [\theta_k(1+\tau_k)]^{-1}[\theta_1\eta_1 P(x^0, x^*) - \tfrac{\theta_k\eta_k}{2}P(x^k, x)] \leq \tfrac{12L_f}{k(k+1)}P(x^0, x^*).$$

■

In view of the results obtained in the above two corollaries, GEM exhibits optimal rates of convergence for both strongly convex and smooth cases. Different from the classical NAG method, GEM performs extrapolation on the gradients, rather than the iterates. This fact will help us to develop an enhanced randomized incremental gradient method than

RPDG in Chapter 2, i.e., the Random Gradient Extrapolation Method, with a much simpler analysis.

3.4 Convergence Analysis of RGEM

Our main goal in this section is to establish the convergence properties of RGEM for solving (3.1.1) and (1.1.5), i.e., the main results stated in Theorem 3.2.1 and 3.2.2. In fact, comparing RGEM in Algorithm 5 with GEM in Algorithm 7, RGEM is a direct randomization of GEM. Therefore, inheriting from GEM, its convergence analysis is carried out completely in the primal space. However, the analysis for RGEM is more challenging especially because we need to 1) build up the relationship between $\frac{1}{m}\sum_{i=1}^m f_i(\underline{x}_i^k)$ and $f(\underline{x}^k)$, for which we exploit the function Q defined in (3.4.3) as an intermediate tool; 2) bound the error caused by inexact gradients at the initial point and 3) analyze the accumulated error caused by randomization and noisy stochastic gradients.

Before proving Theorem 3.2.1 and 3.2.2, we first need to provide some important technical results. Let $\hat{\underline{x}}_i^t$ and $\hat{y}_i^t, i = 1, \ldots, m,\ t \geq 1$ be defined as

$$\hat{\underline{x}}_i^t = (1+\tau_t)^{-1}(x^t + \tau_t \underline{x}_i^{t-1}), \tag{3.4.1}$$

$$\hat{y}_i^t = \begin{cases} \nabla f_i(\hat{\underline{x}}_i^t), & \text{if } y_i^t \text{ is defined in (3.2.9)}, \\ \frac{1}{B_t}\sum_{j=1}^{B_t} G_i(\hat{\underline{x}}_i^t, \xi_{i,j}^t), & \text{if } y_i^t \text{ is defined in (3.2.22)}. \end{cases} \tag{3.4.2}$$

The following simple result demonstrates a few identities related to $\underline{x}_i^t$ (cf. (3.2.8)) and y_i^t (cf. (3.2.9) or (3.2.22)).

Lemma 3.4.7 *Let x^t and y_i^t be defined in (3.2.7) and (3.2.9) (or* (3.2.22)*), respectively, and $\hat{\underline{x}}_i^t$ and $\hat{y}_i^t$ be defined as in* (3.4.1) *and* (3.4.2)*, respectively. Then we have, for any $i = 1, \ldots, m$ and $t = 1, \ldots, k$,*

$$\mathbb{E}_t[y_i^t] = \tfrac{1}{m}\hat{y}_i^t + (1 - \tfrac{1}{m})y_i^{t-1},$$

$$\mathbb{E}_t[\underline{x}_i^t] = \tfrac{1}{m}\hat{\underline{x}}_i^t + (1-\tfrac{1}{m})\underline{x}_i^{t-1},$$
$$\mathbb{E}_t[f_i(\underline{x}_i^t)] = \tfrac{1}{m}f_i(\hat{\underline{x}}_i^t) + (1-\tfrac{1}{m})f_i(\underline{x}_i^{t-1}),$$
$$\mathbb{E}_t[\|\nabla f_i(\underline{x}_i^t) - \nabla f_i(\underline{x}_i^{t-1})\|_*^2] = \tfrac{1}{m}\|\nabla f_i(\hat{\underline{x}}_i^t) - \nabla f_i(\underline{x}_i^{t-1})\|_*^2,$$

where $\mathbb{E}_t$ denotes the conditional expectation w.r.t. i_t given $i_1, \ldots, i_{t-1}$ when y_i^t is defined in (3.2.9)*, and w.r.t. i_t given $i_1, \ldots, i_{t-1}, \xi_1^t, \ldots, \xi_m^t$ when y_i^t is defined in* (3.2.22)*, respectively.*

Proof. This first equality follows immediately from the facts that $\text{Prob}_t\{y_i^t = \hat{y}_i^t\} = \text{Prob}_t\{i_t = i\} = \frac{1}{m}$ and $\text{Prob}_t\{y_i^t = y_i^{t-1}\} = 1 - \frac{1}{m}$. Here Prob_t denotes the conditional probability w.r.t. i_t given $i_1, \ldots, i_{t-1}$ when y_i^t is defined in (3.2.9) and w.r.t i_t given $i_1, \ldots, i_{t-1}, \xi_1^t, \ldots, \xi_m^t$ when y_i^t is defined in (3.2.22), respectively. Similarly, we can prove the rest equalities. ∎

We define the following function Q to help us analyze the convergence properties of RGEM. Let $\underline{x}, x \in X$ be two feasible solutions of (3.1.1) (or (1.1.5)), we define the corresponding $Q(\underline{x}, x)$ by

$$Q(\underline{x}, x) := \langle \nabla f(x), \underline{x} - x\rangle + \mu w(\underline{x}) - \mu w(x). \tag{3.4.3}$$

It is obvious that if we fix $x = x^*$, an optimal solution of (3.1.1) (or (1.1.5)), by the convexity of w and the optimality condition of x^*, for any feasible solution $\underline{x}$, we can conclude that

$$Q(\underline{x}, x^*) \geq \langle \nabla f(x^*) + \mu w'(x^*), \underline{x} - x^*\rangle \geq 0.$$

Moreover, observing that f is smooth, we conclude that

$$Q(\underline{x}, x^*) = f(x^*) + \langle \nabla f(x^*), \underline{x} - x^*\rangle + \mu w(\underline{x}) - \psi(x^*) \geq -\tfrac{L_f}{2}\|\underline{x} - x^*\|^2 + \psi(\underline{x}) - \psi(x^*). \tag{3.4.4}$$

The following lemma establishes an important relationship regarding Q.

Lemma 3.4.8 *Let x^t be defined in* (3.2.7)*, and $x \in X$ be any feasible solution of* (3.1.1) *or* (1.1.5)*. Suppose that τ_t in RGEM satisfy*

$$\theta_t(m(1+\tau_t)-1) = \theta_{t-1}m(1+\tau_{t-1}), \;\; t = 2, \ldots, k, \tag{3.4.5}$$

for some $\theta_t \geq 0, \; t = 1, \ldots, k$. Then, we have

$$\begin{aligned}\textstyle\sum_{t=1}^k \theta_t \mathbb{E}[Q(x^t, x)] \leq \theta_k(1+\tau_k)\textstyle\sum_{i=1}^m \mathbb{E}[f_i(\underline{x}_i^k)] + \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \psi(x)] \\ - \theta_1(m(1+\tau_1)-1)[\langle x^0 - x, \nabla f(x)\rangle + f(x)].\end{aligned} \tag{3.4.6}$$

Proof. In view of the definition of Q in (3.4.3), we have

$$\begin{aligned}Q(x^t, x) &= \tfrac{1}{m}\textstyle\sum_{i=1}^m \langle \nabla f_i(x), x^t - x\rangle + \mu w(x^t) - \mu w(x) \\ &\overset{(3.4.1)}{=} \tfrac{1}{m}\textstyle\sum_{i=1}^m [(1+\tau_t)\langle \hat{\underline{x}}_i^t - x, \nabla f_i(x)\rangle - \tau_t \langle \underline{x}_i^{t-1} - x, \nabla f_i(x)\rangle] + \mu w(x^t) - \mu w(x).\end{aligned}$$

Taking expectation on both sides of the above relation over $\{i_1, \ldots, i_k\}$, and using Lemma 3.4.7, we obtain

$$\begin{aligned}\mathbb{E}[Q(x^t, x)] = \textstyle\sum_{i=1}^m \mathbb{E}[(1+\tau_t)\langle \underline{x}_i^t - x, \nabla f_i(x)\rangle - ((1+\tau_t) - \tfrac{1}{m})\langle \underline{x}_i^{t-1} - x, \nabla f_i(x)\rangle] \\ + \mathbb{E}[\mu w(x^t) - \mu w(x)].\end{aligned}$$

Multiplying both sides of the above inequality by θ_t, and summing up the resulting inequalities from $t = 1$ to k, we conclude that

$$\begin{aligned}\textstyle\sum_{t=1}^k \theta_t \mathbb{E}[Q(x^t, x)] = \textstyle\sum_{i=1}^m \sum_{t=1}^k \mathbb{E}[\theta_t(1+\tau_t)\langle \underline{x}_i^t - x, \nabla f_i(x)\rangle - \theta_t((1+\tau_t) - \tfrac{1}{m})\langle \underline{x}_i^{t-1} - x, \nabla f_i(x)\rangle] \\ + \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \mu w(x)].\end{aligned}$$

Note that by (3.4.5) and the fact that $\underline{x}_i^0 = x^0,\ i = 1, \ldots, m$, we have

$$\sum_{t=1}^k \theta_t = \sum_{t=2}^k [\theta_t m(1+\tau_t) - \theta_{t-1} m(1+\tau_{t-1})] + \theta_1 = \theta_k m(1+\tau_k) - \theta_1(m(1+\tau_1) - 1), \tag{3.4.7}$$

$$\begin{aligned}\sum_{t=1}^k [\theta_t(1+\tau_t)\langle \underline{x}_i^t - x, \nabla f_i(x)\rangle - \theta_t((1+\tau_t) - \tfrac{1}{m})\langle \underline{x}_i^{t-1} - x, \nabla f_i(x)\rangle] \\ = \theta_k(1+\tau_k)\langle \underline{x}_i^k - x, \nabla f_i(x)\rangle - \theta_1((1+\tau_1) - \tfrac{1}{m})\langle x^0 - x, \nabla f_i(x)\rangle, \forall i\end{aligned}$$

Combining the above three relations and using the convexity of f_i, we obtain

$$\begin{aligned}\sum_{t=1}^k \theta_t \mathbb{E}[Q(x^t, x)] \le \theta_k(1+\tau_k)\textstyle\sum_{i=1}^m \mathbb{E}[f_i(\underline{x}_i^k) - f_i(x)] - \theta_1(m(1+\tau_1) - 1)\langle x^0 - x, \nabla f(x)\rangle \\ + \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \mu w(x)],\end{aligned}$$

which in view of (3.4.7) implies (3.4.6). ∎

3.4.1 Convergence Analysis of RGEM for Deterministic Finite-sum Optimization

We now prove the main convergence properties for RGEM to solve (3.1.1). Observe that RGEM starts with $y_i^0 = \mathbf{0},\ i = 1, \ldots, m$, and only updates the corresponding i_t-block of $(\underline{x}_i^t, y_i^t)$, $i = 1, \ldots, m$, according to (3.2.8) and (3.2.9), respectively. Therefore, for y_i^t generated by RGEM, we have

$$y_i^t = \begin{cases} \mathbf{0}, & \text{if the } i\text{-th block has never been updated for the first } t \text{ iterations,} \\ \nabla f_i(\underline{x}_i^t), & \text{o.w.} \end{cases} \tag{3.4.8}$$

Throughout this subsection, we assume that there exists $\sigma_0 \ge 0$ which is the upper bound of the initial gradients, i.e., (3.2.13) holds. Proposition 3.4.9 below establishes some general convergence properties of RGEM for solving strongly convex problems.

Proposition 3.4.9 *Let x^t and $\underline{x}^k$ be defined as in* (3.2.7) *and* (3.2.10)*, respectively, and x^**

be an optimal solution of (3.1.1)*. Let* σ_0 *be defined in* (3.2.13)*, and suppose that* $\{\eta_t\}$*,* $\{\tau_t\}$*, and* $\{\alpha_t\}$ *in RGEM satisfy* (3.4.5) *and*

$$m\theta_{t-1} = \alpha_t\theta_t, \;\; t \geq 2, \tag{3.4.9}$$

$$\theta_t\eta_t \leq \theta_{t-1}(\mu + \eta_{t-1}), \;\; t \geq 2, \tag{3.4.10}$$

$$2\alpha_t L_i \leq m\tau_{t-1}\eta_t, \;\; i = 1, \ldots, m; \; t \geq 2, \tag{3.4.11}$$

$$4L_i \leq \tau_k(\mu + \eta_k), \;\; i = 1, \ldots, m, \tag{3.4.12}$$

for some $\theta_t \geq 0$*,* $t = 1, \ldots, k$*. Then, for any* $k \geq 1$*, we have*

$$\begin{aligned} \mathbb{E}[Q(\underline{x}^k, x^*)] \; &\leq (\textstyle\sum_{t=1}^k \theta_t)^{-1}\tilde{\Delta}_{0,\sigma_0}, \\ \mathbb{E}[P(x^k, x^*)] \; &\leq \tfrac{2\tilde{\Delta}_{0,\sigma_0}}{\theta_k(\mu+\eta_k)}, \end{aligned} \tag{3.4.13}$$

where

$$\tilde{\Delta}_{0,\sigma_0} := \theta_1(m(1+\tau_1)-1)(\psi(x^0)-\psi^*) + \theta_1\eta_1 P(x^0, x^*) + \textstyle\sum_{t=1}^k (\frac{m-1}{m})^{t-1}\frac{2\theta_t\alpha_{t+1}}{m\eta_{t+1}}\sigma_0^2. \tag{3.4.14}$$

Proof. In view of the definition of x^t in (3.2.7) and Lemma 2.5.16, we have

$$\langle x^t - x, \tfrac{1}{m}\textstyle\sum_{i=1}^m \tilde{y}_i^t\rangle + \mu w(x^t) - \mu w(x) \leq \eta_t P(x^{t-1}, x) - (\mu+\eta_t)P(x^t, x) - \eta_t P(x^{t-1}, x^t). \tag{3.4.15}$$

Moreover, using the definition of ψ in (3.1.1), the convexity of f_i, and the fact that $\hat{y}_i^t = \nabla f_i(\hat{\underline{x}}_i^t)$ (see (3.4.2) with y_i^t defined in (3.2.9)), we obtain

$$\begin{aligned} &\tfrac{1+\tau_t}{m}\textstyle\sum_{i=1}^m f_i(\hat{\underline{x}}_i^t) + \mu w(x^t) - \psi(x) \\ &\qquad \leq \tfrac{1+\tau_t}{m}\textstyle\sum_{i=1}^m f_i(\hat{\underline{x}}_i^t) + \mu w(x^t) - \mu w(x) - \tfrac{1}{m}\textstyle\sum_{i=1}^m [f_i(\hat{\underline{x}}_i^t) + \langle \hat{y}_i^t, x - \hat{\underline{x}}_i^t\rangle] \\ &\qquad = \tfrac{\tau_t}{m}\textstyle\sum_{i=1}^m [f_i(\hat{\underline{x}}_i^t) + \langle \hat{y}_i^t, \underline{x}_i^{t-1} - \hat{\underline{x}}_i^t\rangle] + \mu w(x^t) - \mu w(x) - \tfrac{1}{m}\textstyle\sum_{i=1}^m \langle \hat{y}_i^t, x - x^t\rangle \end{aligned}$$

$$\leq -\tfrac{\tau_t}{2m}\textstyle\sum_{i=1}^m \tfrac{1}{L_i}\|\nabla f_i(\hat{\underline{x}}_i^t) - \nabla f_i(\underline{x}_i^{t-1})\|_*^2 + \tfrac{\tau_t}{m}\sum_{i=1}^m f_i(\underline{x}_i^{t-1})$$
$$+ \mu w(x^t) - \mu w(x) - \tfrac{1}{m}\textstyle\sum_{i=1}^m \langle \hat{y}_i^t, x - x^t\rangle, \tag{3.4.16}$$

where the first equality follows from the definition of $\hat{\underline{x}}_i^t$ in (3.4.1), and the last inequality follows from the smoothness of f_i (see Theorem 2.1.5 in [6]) and (3.4.2). It then follows from (3.4.15) and the definition of $\tilde{y}_i^t$ in (3.2.6) that

$$\begin{aligned}\tfrac{1+\tau_t}{m}\textstyle\sum_{i=1}^m f_i(\hat{\underline{x}}_i^t) + \mu w(x^t) - \psi(x) \leq & -\tfrac{\tau_t}{2m}\textstyle\sum_{i=1}^m \tfrac{1}{L_i}\|\nabla f_i(\hat{\underline{x}}_i^t) - \nabla f_i(\underline{x}_i^{t-1})\|_*^2 + \tfrac{\tau_t}{m}\textstyle\sum_{i=1}^m f_i(\underline{x}_i^{t-1}) \\ & + \langle x^t - x, \tfrac{1}{m}\textstyle\sum_{i=1}^m [\hat{y}_i^t - y_i^{t-1} - \alpha_t(y_i^{t-1} - y_i^{t-2})]\rangle \\ & + \eta_t P(x^{t-1}, x) - (\mu + \eta_t)P(x^t, x) - \eta_t P(x^{t-1}, x^t).\end{aligned}$$

Therefore, taking expectation on both sides of the above relation over $\{i_1, \ldots, i_k\}$, and using Lemma 3.4.7, we have

$$\begin{aligned}&\mathbb{E}[(1+\tau_t)\textstyle\sum_{i=1}^m f_i(\underline{x}_i^t) + \mu w(x^t) - \psi(x)] \\ &\leq \mathbb{E}[-\tfrac{\tau_t}{2L_{i_t}}\|\nabla f_{i_t}(\underline{x}_{i_t}^t) - \nabla f_{i_t}(\underline{x}_{i_t}^{t-1})\|_*^2 + \tfrac{1}{m}\textstyle\sum_{i=1}^m (m(1+\tau_t) - 1) f_i(\underline{x}_i^{t-1})] \\ &\quad + \mathbb{E}\{\langle x^t - x, \tfrac{1}{m}\textstyle\sum_{i=1}^m [m(y_i^t - y_i^{t-1}) - \alpha_t(y_i^{t-1} - y_i^{t-2})]\rangle\} \\ &\quad + \mathbb{E}[\eta_t P(x^{t-1}, x) - (\mu + \eta_t)P(x^t, x) - \eta_t P(x^{t-1}, x^t)].\end{aligned}$$

Multiplying both sides of the above inequality by θ_t, and summing up the resulting inequalities from $t = 1$ to k, we obtain

$$\begin{aligned}&\textstyle\sum_{t=1}^k \sum_{i=1}^m \mathbb{E}[\theta_t(1+\tau_t) f_i(\underline{x}_i^t)] + \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \psi(x)] \\ &\qquad \leq \textstyle\sum_{t=1}^k \theta_t \mathbb{E}\left[-\tfrac{\tau_t}{2L_{i_t}}\|\nabla f_{i_t}(\underline{x}_{i_t}^t) - \nabla f_{i_t}(\underline{x}_{i_t}^{t-1})\|_*^2 + \textstyle\sum_{i=1}^m ((1+\tau_t) - \tfrac{1}{m}) f_i(\underline{x}_i^{t-1})\right] \\ &\qquad\quad + \textstyle\sum_{t=1}^k \sum_{i=1}^m \theta_t \mathbb{E}[\langle x^t - x, y_i^t - y_i^{t-1} - \tfrac{\alpha_t}{m}(y_i^{t-1} - y_i^{t-2})\rangle] \\ &\qquad\quad + \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[\eta_t P(x^{t-1}, x) - (\mu + \eta_t)P(x^t, x) - \eta_t P(x^{t-1}, x^t)].\end{aligned} \tag{3.4.17}$$

Now by (3.4.9), and the facts that $y_i^{-1} = y_i^0,\ i = 1, \ldots, m$, and that we only update $y_{i_t}^t$ (see (3.2.9)), we have

$$\begin{aligned}
&\textstyle\sum_{t=1}^k \sum_{i=1}^m \theta_t \langle x^t - x, y_i^t - y_i^{t-1} - \frac{\alpha_t}{m}(y_i^{t-1} - y_i^{t-2})\rangle \\
&= \textstyle\sum_{t=1}^k \theta_t \langle x^t - x, y_{i_t}^t - y_{i_t}^{t-1}\rangle - \frac{\theta_t \alpha_t}{m}\langle x^{t-1} - x, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle - \textstyle\sum_{t=2}^k \frac{\theta_t \alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle \\
&\overset{(3.4.9)}{=} \theta_k \langle x^k - x, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \textstyle\sum_{t=2}^k \frac{\theta_t \alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle.
\end{aligned}$$

Moreover, in view of (3.4.10), (3.4.5), and the fact that $\underline{x}_i^0 = x^0,\ i = 1, \ldots, m$, we obtain

$$\begin{aligned}
&\textstyle\sum_{t=1}^k \theta_t [\eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x)] \overset{(3.4.10)}{\leq} \theta_1 \eta_1 P(x^0, x) - \theta_k(\mu + \eta_k) P(x^k, x), \\
&\textstyle\sum_{t=1}^k \sum_{i=1}^m \theta_t (1 + \tau_t) f_i(\underline{x}_i^t) - \theta_t((1 + \tau_t) - \frac{1}{m}) f_i(\underline{x}_i^{t-1}) \\
&\overset{(3.4.5)}{=} \textstyle\sum_{i=1}^m \theta_k (1 + \tau_k) f_i(\underline{x}_i^k) - \theta_1(m(1 + \tau_1) - 1) f(x^0),
\end{aligned}$$

which together with (3.4.17), (3.4.8) and the fact that $\theta_1 \eta_1 P(x^0, x^1) \geq 0$ imply that

$$\begin{aligned}
&\theta_k(1 + \tau_k) \textstyle\sum_{i=1}^m \mathbb{E}[f_i(\underline{x}_i^k)] + \sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \psi(x)] + \frac{\theta_k(\mu + \eta_k)}{2} \mathbb{E}[P(x^k, x)] \\
&\leq \theta_1(m(1 + \tau_1) - 1) f(x^0) + \theta_1 \eta_1 P(x^0, x) \\
&\quad + \textstyle\sum_{t=2}^k \mathbb{E}\left[-\frac{\theta_t \alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle - \theta_t \eta_t P(x^{t-1}, x^t) - \frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}} \|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2 \right] \\
&\quad + \theta_k \mathbb{E}\left[\langle x^k - x, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \frac{(\mu + \eta_k)}{2} P(x^k, x) - \frac{\tau_k}{2L_{i_k}} \|y_{i_k}^k - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2 \right].
\end{aligned} \tag{3.4.18}$$

By the strong convexity of $P(\cdot, \cdot)$ in (3.1.4), the simple relations that $b\langle u, v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a), \forall a > 0$ and $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we have

$$\begin{aligned}
&\textstyle\sum_{t=2}^k \left[-\frac{\theta_t \alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle - \theta_t \eta_t P(x^{t-1}, x^t) - \frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}} \|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2 \right] \\
&\quad \overset{(3.1.4)}{\leq} \textstyle\sum_{t=2}^k \left[-\frac{\theta_t \alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle - \frac{\theta_t \eta_t}{2}\|x^{t-1} - x^t\|^2 - \frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}} \|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2 \right] \\
&\quad \leq \textstyle\sum_{t=2}^k \left[\frac{\theta_{t-1}\alpha_t}{2m\eta_t} \|y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\|_*^2 - \frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}} \|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2 \right]
\end{aligned}$$

$$\leq \sum_{t=2}^{k}\left[\left(\frac{\theta_{t-1}\alpha_t}{m\eta_t}-\frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\right)\|y_{i_{t-1}}^{t-1}-\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2+\frac{\theta_{t-1}\alpha_t}{m\eta_t}\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})-y_{i_{t-1}}^{t-2}\|_*^2\right]$$

which in view of conditions in (3.4.11) implies that

$$\begin{aligned}&\sum_{t=2}^{k}\left[-\frac{\theta_t\alpha_t}{m}\langle x^t-x^{t-1},y_{i_{t-1}}^{t-1}-y_{i_{t-1}}^{t-2}\rangle-\theta_t\eta_t P(x^{t-1},x^t)-\frac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\|y_{i_{t-1}}^{t-1}-\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2\right]\\&\overset{(3.4.11)}{\leq}\sum_{t=2}^{k}\frac{\theta_{t-1}\alpha_t}{m\eta_t}\left[\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})-y_{i_{t-1}}^{t-2}\|_*^2\right].\end{aligned}$$

Similarly, in view of (3.4.12), we obtain

$$\begin{aligned}&\theta_k\left[\langle x^k-x,y_{i_k}^k-y_{i_k}^{k-1}\rangle-\tfrac{(\mu+\eta_k)}{2}P(x^k,x)-\tfrac{\tau_k}{2L_{i_k}}\|y_{i_k}^k-\nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2\right]\\&\leq\tfrac{2\theta_k}{\mu+\eta_k}\left[\|\nabla f_{i_k}(\underline{x}_{i_k}^{k-1})-y_{i_k}^{k-1}\|_*^2\right]\leq\tfrac{2\theta_k\alpha_{k+1}}{m\eta_{k+1}}\left[\|\nabla f_{i_k}(\underline{x}_{i_k}^{k-1})-y_{i_k}^{k-1}\|_*^2\right],\end{aligned}$$

where the last inequality follows from the fact that $m\eta_{k+1}\leq\alpha_{k+1}(\mu+\eta_k)$ (induced from (3.4.9) and (3.4.10)). Therefore, combing the above three relations, we conclude that

$$\begin{aligned}&\theta_k(1+\tau_k)\textstyle\sum_{i=1}^{m}\mathbb{E}[f_i(\underline{x}_i^k)]+\sum_{t=1}^{k}\theta_t\mathbb{E}[\mu w(x^t)-\psi(x)]+\tfrac{\theta_k(\mu+\eta_k)}{2}\mathbb{E}[P(x^k,x)]\\&\qquad\leq\theta_1(m(1+\tau_1)-1)f(x^0)+\theta_1\eta_1P(x^0,x)+\textstyle\sum_{t=1}^{k}\frac{2\theta_t\alpha_{t+1}}{m\eta_{t+1}}\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})-y_{i_t}^{t-1}\|_*^2].\end{aligned}\tag{3.4.19}$$

We now provide a bound on $\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})-y_{i_t}^{t-1}\|_*^2]$. In view of (3.4.8), we have

$$\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})-y_{i_t}^{t-1}\|_*^2=\begin{cases}\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})\|_*^2, & \text{if the } i_t\text{-th block has never been updated until iteration } t;\\ 0, & \text{o.w.}\end{cases}$$

Let us denote event $\mathcal{B}_{i_t}:=\{\text{the } i_t\text{-th block has never been updated until iteration } t\}$, for all $t=1,\ldots,k$, we have

$$\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})-y_{i_t}^{t-1}\|_*^2]=\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1})\|_*^2|\mathcal{B}_{i_t}]\text{Prob}\{\mathcal{B}_{i_t}\}\leq\left(\tfrac{m-1}{m}\right)^{t-1}\sigma_0^2,$$

where the last inequality follows from the definitions of $\mathcal{B}_{i_t}$, $\underline{x}_i^t$ in (3.2.8) and σ_0^2 in (3.2.13). Fixing $x = x^*$, and using the above result in (3.4.19), we then conclude from (3.4.19) and Lemma 3.4.8 that

$$\begin{aligned} 0 \leq \textstyle\sum_{t=1}^k \theta_t \mathbb{E}[Q(x^t, x^*)] \; &\leq \theta_1(m(1+\tau_1)-1)[f(x^0) - \langle x^0 - x^*, \nabla f(x^*)\rangle - f(x^*)] \\ &\quad + \theta_1\eta_1 P(x^0, x^*) + \textstyle\sum_{t=1}^k (\frac{m-1}{m})^{t-1} \frac{2\theta_t\alpha_{t+1}}{m\eta_{t+1}}\sigma_0^2 - \frac{\theta_k(\mu+\eta_k)}{2}\mathbb{E}[P(x^k, x^*)], \end{aligned}$$

which, in view of the relation $-\langle x^0 - x^*, \nabla f(x^*)\rangle \leq \langle x^0 - x^*, \mu w'(x^*)\rangle \leq \mu w(x^0) - \mu w(x^*)$ and the convexity of $Q(\cdot, x^*)$, implies the first result in (3.4.13). Moreover, we can also conclude from the above inequality that

$$\tfrac{\theta_k(\mu+\eta_k)}{2}\mathbb{E}[P(x^k, x^*)] \; \leq \theta_1(m(1+\tau_1)-1)[\psi(x^0) - \psi(x^*)] + \theta_1\eta_1 P(x^0, x^*) + \textstyle\sum_{t=1}^k (\frac{m-1}{m})^{t-1} \frac{2\theta_t\alpha_{t+1}}{m\eta_{t+1}}\sigma_0^2,$$

from which the second result in (3.4.13) follows. ■

With the help of Proposition 3.4.9, we are now ready to prove Theorem 3.2.1, which establishes the convergence properties of RGEM. In particular, Theorem 3.2.1 shows that RGEM can achieve the optimal convergence rate as $\mathcal{O}\left\{\left(m + \sqrt{m\hat{L}/\mu}\right)\log 1/\epsilon\right\}$ for strongly convex problems.

Proof of Theorem 3.2.1. Letting $\theta_t = \alpha^{-t},\ t = 1, \ldots, k$, we can easily check that parameter setting in (3.2.14) with α defined in (3.2.15) satisfies conditions (3.4.5) and (3.4.9)-(3.4.12) stated in Proposition 3.4.9. It then follows from (3.2.14) and (3.4.13) that

$$\begin{aligned} &\mathbb{E}[Q(\underline{x}^k, x^*)] \leq \tfrac{\alpha^k}{1-\alpha^k}\left[\mu P(x^0, x^*) + \psi(x^0) - \psi^* + \tfrac{2m(1-\alpha)^2\sigma_0^2}{(m-1)\mu}\textstyle\sum_{t=1}^k \left(\frac{m-1}{m\alpha}\right)^t\right], \\ &\mathbb{E}[P(x^k, x^*)] \leq 2\alpha^k\left[P(x^0, x^*) + \tfrac{\psi(x^0)-\psi^*}{\mu} + \tfrac{2m(1-\alpha)^2\sigma_0^2}{(m-1)\mu^2}\textstyle\sum_{t=1}^k \left(\frac{m-1}{m\alpha}\right)^t\right],\ \forall k \geq 1. \end{aligned}$$

Also observe that $\alpha \geq \frac{2m-1}{2m}$, we then have

$$\sum_{t=1}^{k} \left(\frac{m-1}{m\alpha}\right)^t \leq \sum_{t=1}^{k} \left(\frac{2(m-1)}{2m-1}\right)^t \leq 2(m-1).$$

Combining the above three relations and the fact that $m(1-\alpha) \leq 1/2$, we have

$$\begin{aligned} &\mathbb{E}[Q(\underline{x}^k, x^*)] \leq \tfrac{\alpha^k}{1-\alpha^k}\Delta_{0,\sigma_0}, \\ &\mathbb{E}[P(x^k, x^*)] \leq 2\alpha^k \Delta_{0,\sigma_0}/\mu, \ \forall k \geq 1, \end{aligned} \tag{3.4.20}$$

where Δ_{0,σ_0} is defined in (3.2.18). The second relation immediately implies our bound in (3.2.16). Moreover, by the strong convexity of $P(\cdot,\cdot)$ in (3.1.4) and (3.2.16), we have

$$\begin{aligned} \tfrac{L_f}{2}\mathbb{E}[\|\underline{x}^k - x^*\|^2] \ &\leq \tfrac{L_f}{2}(\textstyle\sum_{t=1}^k \theta_t)^{-1}\sum_{t=1}^k \theta_t \mathbb{E}[\|x^t - x^*\|^2] \overset{(3.1.4)}{\leq} L_f \tfrac{(1-\alpha)\alpha^k}{1-\alpha^k}\textstyle\sum_{t=1}^k \alpha^{-t}\mathbb{E}[P(x^t, x^*)] \\ &\overset{(3.2.16)}{\leq} \tfrac{L_f(1-\alpha)\alpha^k}{1-\alpha^k}\textstyle\sum_{t=1}^k \tfrac{2\Delta_{0,\sigma_0}}{\mu} = \tfrac{2L_f(1-\alpha)\Delta_{0,\sigma_0}k\alpha^k}{\mu(1-\alpha^k)}. \end{aligned}$$

Combining the above relation with the first inequality in (3.4.20) and (3.4.4), we obtain

$$\begin{aligned} \mathbb{E}[\psi(\underline{x}^k) - \psi(x^*)] &\overset{(3.4.4)}{\leq} \mathbb{E}[Q(\underline{x}^k, x^*)] + \tfrac{L_f}{2}\mathbb{E}[\|\underline{x}^k - x^*\|^2] \\ &\leq \left(1 + \tfrac{2L_f(1-\alpha)}{\mu}k\right)\tfrac{\Delta_{0,\sigma_0}\alpha^k}{1-\alpha^k} = \left(\tfrac{1}{1-\alpha} + \tfrac{2L_f}{\mu}k\right)\tfrac{\Delta_{0,\sigma_0}\alpha^k(1-\alpha)}{1-\alpha^k}. \end{aligned}$$

Observing that

$$\begin{aligned} \tfrac{1}{1-\alpha} &\leq \tfrac{16}{3}\max\{m, \hat{L}/\mu\}, \\ \tfrac{2L_f}{\mu} &\leq \tfrac{16}{3}\max\{m, \hat{L}/\mu\}, \\ (k+1)\tfrac{\alpha^k(1-\alpha)}{1-\alpha^k} &= \left(\textstyle\sum_{t=1}^k \tfrac{\alpha^t}{\alpha^t} + 1\right)\tfrac{\alpha^k(1-\alpha)}{1-\alpha^k} \leq \left(\textstyle\sum_{t=1}^k \tfrac{\alpha^t}{\alpha^{3t/2}} + 1\right)\tfrac{\alpha^k(1-\alpha)}{1-\alpha^k} \\ &\leq \tfrac{1-\alpha^{k/2}}{\alpha^{k/2}(1-\alpha^{1/2})}\tfrac{\alpha^k(1-\alpha)}{1-\alpha^k} + \alpha^k \leq 2\alpha^{k/2} + \alpha^k \leq 3\alpha^{k/2}, \end{aligned}$$

we have

$$\mathbb{E}[\psi(\underline{x}^k) - \psi(x^*)] \leq \tfrac{16}{3} \max\left\{m, \tfrac{\hat{L}}{\mu}\right\} \tfrac{(k+1)\alpha^k(1-\alpha)\Delta_{0,\sigma_0}}{1-\alpha^k} \leq 16 \max\left\{m, \tfrac{\hat{L}}{\mu}\right\} \Delta_{0,\sigma_0}\alpha^{k/2}.$$

3.4.2 Convergence Analysis of RGEM for Stochastic Finite-sum Optimization

Our goal in this section is to establish the convergence properties of RGEM for solving stochastic finite-sum optimization problems in (1.1.5). For notation convenience, we use $\mathbb{E}_{[i_k]}$ for taking expectation over $\{i_1, \ldots, i_k\}$, $\mathbb{E}_\xi$ for expectations over $\{\xi^1, \ldots, \xi^k\}$, respectively, we use $\mathbb{E}$ to denote the expectations over all random variables.

Note that the parameter $\{B_t\}$ in Algorithm 6 denotes the batch size used to compute $y_{i_t}^t$ in (3.2.22). Since we now assume that $\|\cdot\|$ is associated with a certain inner product, it can be easily seen from (3.2.22), and the two assumptions we have for the stochastic gradients computed by $\mathcal{SFO}$ oracle, i.e., (3.2.20) and (3.2.21), that

$$\mathbb{E}_\xi[y_{i_t}^t] = \nabla f_{i_t}(\underline{x}_{i_t}^t) \text{ and } \mathbb{E}_\xi[\|y_{i_t}^t - \nabla f_{i_t}(\underline{x}_{i_t}^t)\|_*^2] \leq \tfrac{\sigma^2}{B_t},\ \forall i_t, t = 1, \ldots, k, \tag{3.4.21}$$

and hence $y_{i_t}^t$ is an unbiased estimator of $\nabla f_{i_t}(\underline{x}_{i_t}^t)$. Moreover, for y_i^t generated by Algorithm 6, we can see that

$$y_i^t = \begin{cases} \mathbf{0}, & \text{if the } i\text{-th block has never been updated for the first } t \text{ iterations;} \\ \frac{1}{B_l}\sum_{j=1}^{B_l} G_i(\underline{x}_i^l, \xi_{i,j}^l), & \text{if the latest update happened at } l\text{-th iteration, for } 1 \leq l \leq t. \end{cases} \tag{3.4.22}$$

We first establish some general convergence properties for Algorithm 6.

Proposition 3.4.10 *Let x^t and $\underline{x}^k$ be defined as in* (3.2.7) *and* (3.2.10)*, respectively, and x^* be an optimal solution of* (1.1.5)*. Suppose that σ_0 and σ are defined in* (3.2.13) *and* (3.2.21)*, respectively, and $\{\eta_t\}$, $\{\tau_t\}$, and $\{\alpha_t\}$ in Algorithm 6 satisfy* (3.4.5)*,* (3.4.9)

(3.4.10), *and* (3.4.12) *for some* $\theta_t \geq 0$, $t = 1, \ldots, k$. *Moreover, if*

$$3\alpha_t L_i \leq m\tau_{t-1}\eta_t,\ i = 1, \ldots, m; t \geq 2, \tag{3.4.23}$$

then for any $k \geq 1$, *we have*

$$\begin{aligned} \mathbb{E}[Q(\underline{x}^k, x^*)] &\leq (\textstyle\sum_{t=1}^k \theta_t)^{-1} \tilde{\Delta}_{0,\sigma_0,\sigma}, \\ \mathbb{E}[P(x^k, x^*)] &\leq \tfrac{2\tilde{\Delta}_{0,\sigma_0,\sigma}}{\theta_k(\mu+\eta_k)}, \end{aligned} \tag{3.4.24}$$

where

$$\tilde{\Delta}_{0,\sigma_0,\sigma} := \tilde{\Delta}_{0,\sigma_0} + \textstyle\sum_{t=2}^k \frac{3\theta_{t-1}\alpha_t\sigma^2}{2m\eta_t B_{t-1}} + \sum_{t=1}^k \frac{2\theta_t\alpha_{t+1}}{m^2\eta_{t+1}} \sum_{l=1}^{t-1} (\frac{m-1}{m})^{t-1-l} \frac{\sigma^2}{B_l}, \tag{3.4.25}$$

with $\tilde{\Delta}_{0,\sigma_0}$ *defined in* (3.4.14).

Proof. Observe that in Algorithm 6 y_i^t is updated as in (3.2.22). Therefore, according to (3.4.2), we have

$$\hat{y}_i^t = \tfrac{1}{B_t} \textstyle\sum_{j=1}^{B_t} G_i(\underline{\hat{x}}_i^t, \xi_{i,j}^t),\ i = 1, \ldots, m,\ t \geq 1,$$

which together with the first relation in (3.4.21) imply that $\mathbb{E}_\xi[\langle \hat{y}_i^t, x - \underline{\hat{x}}_i^t \rangle] = \mathbb{E}_\xi[\langle \nabla f_i(\underline{\hat{x}}_i^t), x - \underline{\hat{x}}_i^t \rangle]$. Hence, we can rewrite (3.4.16) as

$$\begin{aligned} &\mathbb{E}_\xi[\tfrac{1+\tau_t}{m} \textstyle\sum_{i=1}^m f_i(\underline{\hat{x}}_i^t) + \mu w(x^t) - \psi(x)] \\ &\leq \mathbb{E}_\xi\left[\tfrac{1+\tau_t}{m} \textstyle\sum_{i=1}^m f_i(\underline{\hat{x}}_i^t) + \mu w(x^t) - \mu w(x) - \tfrac{1}{m} \textstyle\sum_{i=1}^m [f_i(\underline{\hat{x}}_i^t) + \langle \nabla f_i(\underline{\hat{x}}_i^t), x - \underline{\hat{x}}_i^t \rangle]\right] \\ &= \mathbb{E}_\xi\left[\tfrac{1+\tau_t}{m} \textstyle\sum_{i=1}^m f_i(\underline{\hat{x}}_i^t) + \mu w(x^t) - \mu w(x) - \tfrac{1}{m} \textstyle\sum_{i=1}^m [f_i(\underline{\hat{x}}_i^t) + \langle \hat{y}_i^t, x - \underline{\hat{x}}_i^t \rangle]\right] \\ &\leq \mathbb{E}_\xi\big[- \tfrac{\tau_t}{2m} \textstyle\sum_{i=1}^m \tfrac{1}{L_i} \|\nabla f_i(\underline{\hat{x}}_i^t) - \nabla f_i(\underline{x}_i^{t-1})\|_*^2 + \tfrac{\tau_t}{m} \textstyle\sum_{i=1}^m f_i(\underline{x}_i^{t-1}) \\ &\quad + \langle x^t - x, \tfrac{1}{m} \textstyle\sum_{i=1}^m [\hat{y}_i^t - y_i^{t-1} - \alpha_t(y_i^{t-1} - y_i^{t-2})] \rangle \\ &\quad + \eta_t P(x^{t-1}, x) - (\mu + \eta_t) P(x^t, x) - \eta_t P(x^{t-1}, x^t)\big], \end{aligned}$$

where the last inequality follows from (3.4.15). Following the same procedure as in the proof of Proposition 3.4.9, we obtain the following similar relation (cf. (3.4.18))

$$\begin{aligned}
&\theta_k(1+\tau_k)\textstyle\sum_{i=1}^m \mathbb{E}[f_i(\underline{x}_i^k)] + \sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \psi(x)] + \frac{\theta_k(\mu+\eta_k)}{2}\mathbb{E}[P(x^k, x)] \\
&\quad \le \theta_1(m(1+\tau_1)-1)f(x^0) + \theta_1\eta_1 P(x^0, x) + \textstyle\sum_{t=2}^k \mathbb{E}\Big[-\frac{\theta_t\alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle \\
&\quad - \theta_t\eta_t P(x^{t-1}, x^t) - \tfrac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2\Big] \\
&\quad + \theta_k \mathbb{E}\left[\langle x^k - x, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \tfrac{(\mu+\eta_k)}{2}P(x^k, x) - \tfrac{\tau_k}{2L_{i_k}}\|\nabla f_{i_k}(\underline{x}_{i_k}^k) - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2\right].
\end{aligned}$$

By the strong convexity of $P(\cdot,\cdot)$ in (3.1.4), and the fact that $b\langle u, v\rangle - a\|v\|^2/2 \le b^2\|u\|^2/(2a), \forall a > 0$, we have, for $t = 2, \ldots, k$,

$$\begin{aligned}
&\mathbb{E}[-\tfrac{\theta_t\alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - y_{i_{t-1}}^{t-2}\rangle - \theta_t\eta_t P(x^{t-1}, x^t) - \tfrac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2] \\
&\overset{(3.1.4)}{\le} \mathbb{E}[-\tfrac{\theta_t\alpha_t}{m}\langle x^t - x^{t-1}, y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) + \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) + \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - y_{i_{t-1}}^{t-2}\rangle] \\
&\quad - \mathbb{E}\left[\tfrac{\theta_t\eta_t}{2}\|x^{t-1} - x^t\|^2 + \tfrac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2\right] \\
&\le \mathbb{E}\left[\left(\tfrac{3\theta_{t-1}\alpha_t}{2m\eta_t} - \tfrac{\theta_{t-1}\tau_{t-1}}{2L_{i_{t-1}}}\right)\|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1}) - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2})\|_*^2\right] \\
&\quad + \tfrac{3\theta_{t-1}\alpha_t}{2m\eta_t}\mathbb{E}\left[\|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1})\|_*^2 + \|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - y_{i_{t-1}}^{t-2}\|_*^2\right] \\
&\overset{(3.4.23)}{\le} \tfrac{3\theta_{t-1}\alpha_t}{2m\eta_t}\mathbb{E}\left[\|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1})\|_*^2 + \|\nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-2}) - y_{i_{t-1}}^{t-2}\|_*^2\right].
\end{aligned}$$

Similarly, we can also obtain

$$\begin{aligned}
&\mathbb{E}\left[\langle x^k - x, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \tfrac{(\mu+\eta_k)}{2}P(x^k, x) - \tfrac{\tau_k}{2L_{i_k}}\|f_{i_k}(\underline{x}_{i_k}^k) - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2\right] \\
&\overset{(3.4.21),(3.1.4)}{\le} \mathbb{E}\left[\langle x^k - x, \nabla f_{i_k}(\underline{x}_{i_k}^k) - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1}) + \nabla f_{i_k}(\underline{x}_{i_k}^{k-1}) - y_{i_k}^{k-1}\rangle\right] \\
&\quad - \mathbb{E}\left[\tfrac{(\mu+\eta_k)}{4}\|x^k - x\|^2 + \tfrac{\tau_k}{2L_{i_k}}\|f_{i_k}(\underline{x}_{i_k}^k) - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2\right] \\
&\le \mathbb{E}\left[\left(\tfrac{2}{\mu+\eta_k} - \tfrac{\tau_k}{2L_{i_k}}\right)\|\nabla f_{i_k}(\underline{x}_{i_k}^k) - \nabla f_{i_k}(\underline{x}_{i_k}^{k-1})\|_*^2 + \tfrac{2}{\mu+\eta_k}\|\nabla f_{i_k}(\underline{x}_{i_k}^{k-1}) - y_{i_k}^{k-1}\|_*^2\right] \\
&\overset{(3.4.12)}{\le} \mathbb{E}\left[\tfrac{2}{\mu+\eta_k}\|\nabla f_{i_k}(\underline{x}_{i_k}^{k-1}) - y_{i_k}^{k-1}\|_*^2\right].
\end{aligned}$$

Combining the above three relations, and using the fact that $m\eta_{k+1} \leq \alpha_{k+1}(\mu + \eta_k)$ (induced from (3.4.9) and (3.4.10)), we have

$$\begin{aligned}
&\theta_k(1+\tau_k)\textstyle\sum_{i=1}^m \mathbb{E}[f_i(\underline{x}_i^k)] + \sum_{t=1}^k \theta_t \mathbb{E}[\mu w(x^t) - \psi(x)] + \frac{\theta_k(\mu+\eta_k)}{2}\mathbb{E}[P(x^k, x)] \\
&\quad \leq \theta_1(m(1+\tau_1)-1)f(x^0) + \theta_1\eta_1 P(x^0, x) \\
&\qquad + \textstyle\sum_{t=2}^k \frac{3\theta_{t-1}\alpha_t}{2m\eta_t}\mathbb{E}[\|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1})\|_*^2] + \sum_{t=1}^k \frac{2\theta_t\alpha_{t+1}}{m\eta_{t+1}}\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1}) - y_{i_t}^{t-1}\|_*^2].
\end{aligned}$$

Moreover, in view of the second relation in (3.4.21), we have

$$\mathbb{E}[\|y_{i_{t-1}}^{t-1} - \nabla f_{i_{t-1}}(\underline{x}_{i_{t-1}}^{t-1})\|_*^2] \leq \tfrac{\sigma^2}{B_{t-1}},\ \forall t \geq 2.$$

Let us denote $\mathcal{E}_{i_t,t} := \max\{l : i_l = i_t, l < t\}$ with $\mathcal{E}_{i_t,t} = 0$ denoting the event that the i_t-th block has never been updated until iteration t, we can also conclude that for any $t \geq 1$

$$\begin{aligned}
\mathbb{E}[\|\nabla f_{i_t}(\underline{x}_{i_t}^{t-1}) - y_{i_t}^{t-1}\|_*^2] &= \textstyle\sum_{l=0}^{t-1}\mathbb{E}\left[\|\nabla f_{i_l}(\underline{x}_{i_l}^{l}) - y_{i_l}^{l}\|_*^2 | \{\mathcal{E}_{i_t,t} = l\}\right] \mathrm{Prob}\{\mathcal{E}_{i_t,t} = l\} \\
&\leq (\tfrac{m-1}{m})^{t-1}\sigma_0^2 + \textstyle\sum_{l=1}^{t-1}\frac{1}{m}(\frac{m-1}{m})^{t-1-l}\frac{\sigma^2}{B_l},
\end{aligned}$$

where the first term in the inequality corresponds to the case when the i_t-block has never been updated for the first $t-1$ iterations, and the second term represents that its latest update for the first $t-1$ iterations happened at the l-th iteration. Hence, using Lemma 3.4.8 and following the same argument as in the proof of Proposition 3.4.9, we obtain our results in (3.4.24). ∎

We are now ready to prove Theorem 3.2.2, which establishes an optimal complexity bound (up to a logarithmic factor) on the number of calls to the $\mathcal{SFO}$ oracle and a linear rate of convergence in terms of the communication complexity for solving problem (1.1.5).

Proof of Theorem 3.2.2 Let us set $\theta_t = \alpha^{-t},\ t = 1, \ldots, k$. It is easy to check that the parameter setting in (3.2.14) with α defined in (3.2.15) satisfies conditions (3.4.5), (3.4.9),

(3.4.10), (3.4.12), and (3.4.23) as required by Proposition 3.4.10. By (3.2.14), the definition of B_t in (3.2.23), and the fact that $\alpha \geq \frac{2m-1}{2m} > (m-1)/m$, we have

$$\begin{aligned}
\textstyle\sum_{t=2}^{k} \frac{3\theta_{t-1}\alpha_t\sigma^2}{2m\eta_t B_{t-1}} &\leq \textstyle\sum_{t=2}^{k} \frac{3\sigma^2}{2\mu(1-\alpha)k} \leq \frac{3\sigma^2}{2\mu(1-\alpha)}, \\
\textstyle\sum_{t=1}^{k} \frac{2\theta_t\alpha_{t+1}}{m^2\eta_{t+1}} \textstyle\sum_{l=1}^{t-1} (\frac{m-1}{m})^{t-1-l} \frac{\sigma^2}{B_l} &\leq \frac{2\sigma^2}{\alpha\mu m(1-\alpha)k} \textstyle\sum_{t=1}^{k} (\frac{m-1}{m\alpha})^{t-1} \sum_{l=1}^{t-1} (\frac{m\alpha}{m-1})^{l} \\
&\leq \frac{2\sigma^2}{\mu(1-\alpha)m\alpha k} \textstyle\sum_{t=1}^{k} (\frac{m-1}{m\alpha})^{t-1} (\frac{m\alpha}{m-1})^{t-1} \frac{1}{1-(m-1)/(m\alpha)} \\
&\leq \frac{2\sigma^2}{\mu(1-\alpha)} \frac{1}{m\alpha-(m-1)} \leq \frac{4\sigma^2}{\mu(1-\alpha)}.
\end{aligned}$$

Hence, similar to the proof of Theorem 3.2.1, using the above relations and (3.2.14) in (3.4.24), we obtain

$$\begin{aligned}
\mathbb{E}[Q(\underline{x}^k, x^*)] &\leq \tfrac{\alpha^k}{1-\alpha^k} \left[\Delta_{0,\sigma_0} + \tfrac{5\sigma^2}{\mu} \right], \\
\mathbb{E}[P(x^k, x^*)] &\leq 2\alpha^k \left[\Delta_{0,\sigma_0} + \tfrac{5\sigma^2}{\mu^2} \right],
\end{aligned}$$

where Δ_{0,σ_0} is defined in (3.2.18). The second relation implies our results in (3.2.24). Moreover, (3.2.25) follows from the same argument as we used in proving Theorem 3.2.1.

3.5 Concluding Remarks of This Chapter

In this chapter, we propose a new randomized incremental gradient method, referred to as random gradient extrapolation method, for solving the classes of deterministic finite-sum optimization problems in (3.1.1) and stochastic finite-sum optimization problems in (1.1.5), respectively. We demonstrate that without any exact gradient evaluation even at the initial point, this algorithm achieves optimal linear rate of convergence for deterministic strongly convex problems, as well as exhibiting optimal sublinear rate of convergence (up to a logarithmic factor) for stochastic strongly convex problems. All these complexity bounds have been established in terms of the total number of gradient computations of component function f_i and the latter complexity bound on the computation of stochastic gradients is in

fact asymptotically independent of the number of components m. Moreover, we consider solving finite-sum problems in (3.1.1) and (1.1.5) in a distributed network setting with m agents connected to a central server. Since each iteration of our proposed algorithm only involves constant number of communication rounds between the server and one randomly selected agent, it achieves linear communication complexity and avoids synchronous delays among agents. It is worth pointing out that by exploiting the mini-batch technique, the algorithm can also achieve linear communication complexity for solving stochastic finite-sum problems which is the best-known communication complexity bound for distributed stochastic optimization problems in the literature.

CHAPTER 4
COMMUNICATION-EFFICIENT ALGORITHMS FOR DECENTRALIZED AND STOCHASTIC OPTIMIZATION

4.1 Overview

We present a new class of decentralized first-order methods for nonsmooth and stochastic optimization problems defined over multiagent networks. Considering that communication is a major bottleneck in decentralized optimization, our main goal in this chapter is to develop algorithmic frameworks which can significantly reduce the number of inter-node communications. Our major contribution is to present a new class of decentralized primal-dual type algorithms, namely the decentralized communication sliding (DCS) methods, which can skip the inter-node communications while agents solve the primal subproblems iteratively through linearizations of their local objective functions. By employing DCS, agents can find an ϵ-solution both in terms of functional optimality gap and feasibility residual in $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication rounds for general convex functions (resp., strongly convex functions), while maintaining the $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) bound on the total number of intra-node subgradient evaluations. We also present a stochastic counterpart for these algorithms, denoted by SDCS, for solving stochastic optimization problems whose objective function cannot be evaluated exactly. In comparison with existing results for decentralized nonsmooth and stochastic optimization, we can reduce the total number of inter-node communication rounds by orders of magnitude while still maintaining the optimal complexity bounds on intra-node stochastic subgradient evaluations. The bounds on the (stochastic) subgradient evaluations are actually comparable to those required for centralized nonsmooth and stochastic optimization under certain conditions on the target accuracy.

To the best of our knowledge, this is the first time that these communication sliding algorithms, and the aforementioned separate complexity bounds on communication rounds and (stochastic) subgradient evaluations are presented in the literature. Table 4.1 summarizes the improvement on communication complexity obtained by our algorithms over existing methods for decentralized nonsmooth and stochastic optimization.

Table 4.1: Summary of communication complexities for obtaining a (stochastic) ϵ-solution of (1.2.13)

Problem type: f_i	**Communication rounds**	
	Our results	Existing results
Deterministic, convex	$\mathcal{O}\{1/\epsilon\}$	$\mathcal{O}\{1/\epsilon^2\}$
Deterministic, strongly convex	$\mathcal{O}\{1/\sqrt{\epsilon}\}$	$\mathcal{O}\{1/\epsilon\}$
Stochastic, convex	$\mathcal{O}\{1/\epsilon\}$	$\mathcal{O}\{1/\epsilon^2\}$
Stochastic, strongly convex	$\mathcal{O}\{1/\sqrt{\epsilon}\}$	$\mathcal{O}\{1/\epsilon\}$

This chapter is organized as follows. In Section 4.2, we introduce the problem formulation and the definition of the gap function, which will be used as the termination criterion of our methods. We also provide some preliminaries on distance generating functions and prox-functions. In Section 4.3, we present the communication sliding algorithms when the exact subgradients of f_i's are available and establish their convergence properties for the general and strongly convex cases. In Section 4.4, we generalize the algorithm in Section 4.3 for stochastic optimization problems. The proofs of some important technical results in Section 4.3 and 4.4 are provided in Section 4.5. We also provide some preliminary numerical results in Section 4.6 to demonstrate the advantages of our algorithms. Some concluding remarks are made in Section 4.7.

4.1.1 Notation and Terminologies

Let $\mathbb{R}$ denote the set of real numbers. All vectors are viewed as column vectors, and for a vector $x \in \mathbb{R}^d$, we use $x^\top$ to denote its transpose. For a stacked vector of x_i's, we often use $(x_1, \ldots, x_m)$ to represent the column vector $[x_1^\top, \ldots, x_m^\top]^\top$. We denote by $\mathbf{0}$ and $\mathbf{1}$ the vector of all zeros and ones whose dimensions vary from the context. The cardinality of a set S is denoted by $|S|$. We use I_d to denote the identity matrix in $\mathbb{R}^{d\times d}$. We use $A \otimes B$ for matrices $A \in \mathbb{R}^{n_1\times n_2}$ and $B \in \mathbb{R}^{m_1\times m_2}$ to denote their Kronecker product of size $\mathbb{R}^{n_1m_1\times n_2m_2}$. For a matrix $A \in \mathbb{R}^{n\times m}$, we use A_{ij} to denote the entry of i-th row and j-th column. For any $m \geq 1$, the set of integers $\{1, \ldots, m\}$ is denoted by $[m]$.

4.2 Preliminaries

In Subsections 4.2.1 we introduce the saddle point reformulation of (1.2.13) and define appropriate gap functions which will be used for the convergence analysis of our algorithms. Moreover, in Subsection 4.2.2 we provide a brief review on the distance generating function and prox-function.

4.2.1 Problem Formulation and Termination Criteria

We consider a reformulation of the problem (1.2.19) (equivalently (1.2.13)) as a saddle point problem. By the method of Lagrange multipliers, problem (1.2.19) is equivalent to the following saddle point problem:

$$\min_{\mathbf{x}\in X^m} \left[F(\mathbf{x}) + \max_{\mathbf{y}\in\mathbb{R}^{md}} \langle \mathbf{L}\mathbf{x}, \mathbf{y}\rangle \right], \tag{4.2.1}$$

where $X^m := X_1 \times \ldots \times X_m$ and $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{R}^{md}$ are the Lagrange multipliers associated with the constraints $\mathbf{L}\mathbf{x} = \mathbf{0}$. We assume that there exists an optimal solution $\mathbf{x}^* \in X^m$ of (1.2.19) and that there exists $\mathbf{y}^* \in \mathbb{R}^{md}$ such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a saddle point of (4.2.1). In fact, since our objective function $F(\mathbf{x})$ is convex, strong duality holds if

constraint qualification (CQ) condition holds. In particular, CQ condition states that there exists $\bar{\mathbf{x}} \in X^m$ such that $\mathbf{L}\bar{\mathbf{x}} = \mathbf{0}$, which is implied by the assumption that there exists an optimal solution to (1.2.19).

Given a pair of feasible solutions $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ and $\bar{\mathbf{z}} = (\bar{\mathbf{x}}, \bar{\mathbf{y}})$ of (4.2.1), we define the *primal-dual gap function* $Q(\mathbf{z}; \bar{\mathbf{z}})$ by

$$Q(\mathbf{z}; \bar{\mathbf{z}}) := F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}, \bar{\mathbf{y}} \rangle - [F(\bar{\mathbf{x}}) + \langle \mathbf{L}\bar{\mathbf{x}}, \mathbf{y} \rangle]. \tag{4.2.2}$$

Sometimes we also use the notations $Q(\mathbf{z}; \bar{\mathbf{z}}) := Q(\mathbf{x}, \mathbf{y}; \bar{\mathbf{x}}, \bar{\mathbf{y}})$ or $Q(\mathbf{z}; \bar{\mathbf{z}}) := Q(\mathbf{x}, \mathbf{y}; \bar{\mathbf{z}}) = Q(\mathbf{z}; \bar{\mathbf{x}}, \bar{\mathbf{y}})$. One can easily see that $Q(\mathbf{z}^*; \mathbf{z}) \leq 0$ and $Q(\mathbf{z}; \mathbf{z}^*) \geq 0$ for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$, where $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ is a saddle point of (4.2.1). For compact sets $X^m \subset \mathbb{R}^{md}$, $Y \subset \mathbb{R}^{md}$, the gap function

$$\sup_{\bar{\mathbf{z}} \in X^m \times Y} Q(\mathbf{z}; \bar{\mathbf{z}}) \tag{4.2.3}$$

measures the accuracy of the approximate solution $\mathbf{z}$ to the saddle point problem (4.2.1).

However, the saddle point formulation (4.2.1) of our problem of interest (1.2.13) may have an unbounded feasible set. We adopt the perturbation-based termination criterion by Monteiro and Svaiter [101, 102, 103] and propose a modified version of the gap function in (4.2.3). More specifically, we define

$$g_Y(\mathbf{s}, \mathbf{z}) := \sup_{\bar{\mathbf{y}} \in Y} Q(\mathbf{z}; \mathbf{x}^*, \bar{\mathbf{y}}) - \langle \mathbf{s}, \bar{\mathbf{y}} \rangle, \tag{4.2.4}$$

for any closed set $Y \subseteq \mathbb{R}^{md}$, $\mathbf{z} \in X^m \times \mathbb{R}^{md}$ and $\mathbf{s} \in \mathbb{R}^{md}$. If $Y = \mathbb{R}^{md}$, we omit the subscript Y and simply use the notation $g(\mathbf{s}, \mathbf{z})$.

This perturbed gap function allows us to bound the objective function value and the feasibility separately. We first define the following terminology.

Definition 1 *A point* $\mathbf{x} \in X^m$ *is called an* (ϵ, δ)*-solution of* (1.2.19) *if*

$$F(\mathbf{x}) - F(\mathbf{x}^*) \leq \epsilon \text{ and } \|\mathbf{L}\mathbf{x}\| \leq \delta. \tag{4.2.5}$$

We say that $\mathbf{x}$ *has primal residual* ϵ *and feasibility residual* δ*.*

Similarly, a stochastic (ϵ, δ)-solution of (1.2.19) can be defined as a point $\hat{\mathbf{x}} \in X^m$ s.t. $\mathbb{E}[F(\hat{\mathbf{x}}) - F(\mathbf{x}^*)] \leq \epsilon$ and $\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}\|] \leq \delta$ for some $\epsilon, \delta > 0$. Note that for problem (1.2.19), the feasibility residual measures the disagreement among the local copies x_i, for $i \in \mathcal{N}$.

In the following proposition, we adopt a result from [104, Proposition 2.1] to describe the relationship between the perturbed gap function (4.2.4) and the approximate solutions to problem (1.2.19). Although the proposition was originally developed for deterministic cases, the extension of this to stochastic cases is straightforward.

Proposition 4.2.1 *For any* $Y \subset \mathbb{R}^{md}$ *such that* $\mathbf{0} \in Y$*, if* $g_Y(\mathbf{L}\mathbf{x}, \mathbf{z}) \leq \epsilon < \infty$ *and* $\|\mathbf{L}\mathbf{x}\| \leq \delta$*, where* $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$*, then* $\mathbf{x}$ *is an* (ϵ, δ)*-solution of* (1.2.19)*. In particular, when* $Y = \mathbb{R}^{md}$*, for any* $\mathbf{s}$ *such that* $g(\mathbf{s}, \mathbf{z}) \leq \epsilon < \infty$ *and* $\|\mathbf{s}\| \leq \delta$*, we always have* $\mathbf{s} = \mathbf{L}\mathbf{x}$*.*

4.2.2 Distance Generating Function and Prox-function

In this subsection, we define the concept of prox-function, which is also known as proximity control function or Bregman distance function [86]. Prox-function has played an important role in the recent development of first-order methods for convex programming as a substantial generalization of the Euclidean projection. Unlike the standard projection operator $\Pi_U[x] := \operatorname{argmin}_{u \in U} \|x - u\|^2$, which is inevitably tied to the Euclidean geometry, prox-function can be flexibly tailored to the geometry of a constraint set U.

For any convex set U equipped with an arbitrary norm $\|\cdot\|_U$, we say that a function $\omega : U \to \mathbb{R}$ is a *distance generating function* with modulus $\nu > 0$ with respect to $\|\cdot\|_U$, if ω is continuously differentiable and strongly convex with modulus ν with respect to $\|\cdot\|_U$,

i.e.,

$$\langle \nabla\omega(x) - \nabla\omega(u), x - u\rangle \geq \nu\|x-u\|_U^2, \quad \forall x, u \in U. \tag{4.2.6}$$

The *prox-function*, or *Bregman distance function*, induced by ω is given by

$$V(x,u) \equiv V_\omega(x,u) := \omega(u) - [\omega(x) + \langle \nabla\omega(x), u - x\rangle]. \tag{4.2.7}$$

It then follows from the strong convexity of ω that

$$V(x,u) \geq \tfrac{\nu}{2}\|x-u\|_U^2, \quad \forall x, u \in U.$$

We now assume that the individual constraint set X_i for each agent in problem (1.2.13) are equipped with norm $\|\cdot\|_{X_i}$, and their associated prox-functions are given by $V_i(\cdot,\cdot)$. Moreover, we assume that each $V_i(\cdot,\cdot)$ shares the same strongly convex modulus $\nu = 1$, i.e.,

$$V_i(x_i,u_i) \geq \tfrac{1}{2}\|x_i - u_i\|_{X_i}^2, \quad \forall x_i, u_i \in X_i,\ i = 1,\ldots,m. \tag{4.2.8}$$

We define the norm associated with the primal feasible set $X^m = X_1 \times \ldots \times X_m$ of (4.2.1) as follows:[1]

$$\|\mathbf{x}\|^2 \equiv \|\mathbf{x}\|_{X^m}^2 := \textstyle\sum_{i=1}^m \|x_i\|_{X_i}^2, \tag{4.2.9}$$

where $\mathbf{x} = (x_1,\ldots,x_m) \in X^m$ for any $x_i \in X_i$. Therefore, the corresponding prox-function $\mathbf{V}(\cdot,\cdot)$ can be defined as

$$\mathbf{V}(\mathbf{x},\mathbf{u}) := \textstyle\sum_{i=1}^m V_i(x_i,u_i),\ \forall \mathbf{x}, \mathbf{u} \in X^m. \tag{4.2.10}$$

[1] We can define the norm associated with X^m in a more general way, e.g., $\|\mathbf{x}\|^2 := \sum_{i=1}^m p_i\|x_i\|_{X_i}^2$, $\forall \mathbf{x} = (x_1,\ldots,x_m) \in X^m$, for some $p_i > 0$, $i = 1,\ldots,m$. Accordingly, the prox-function $\mathbf{V}(\cdot,\cdot)$ can be defined as $\mathbf{V}(\mathbf{x},\mathbf{u}) := \sum_{i=1}^m p_i V_i(x_i,u_i)$, $\forall \mathbf{x}, \mathbf{u} \in X^m$. This setting gives us flexibility to choose p_i's based on the information of individual X_i's, and the possibility to further refine the convergence results.

Note that by (4.2.8) and (4.2.9), it can be easily seen that

$$\mathbf{V}(\mathbf{x}, \mathbf{u}) \geq \tfrac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2, \ \forall \mathbf{x}, \mathbf{u} \in X^m. \tag{4.2.11}$$

Throughout the chapter, we endow the dual space where the multipliers $\mathbf{y}$ of (4.2.1) reside with the standard Euclidean norm $\|\cdot\|_2$, since the feasible region of $\mathbf{y}$ is unbounded. For simplicity, we often write $\|\mathbf{y}\|$ instead of $\|\mathbf{y}\|_2$ for a dual multiplier $\mathbf{y} \in \mathbb{R}^{md}$.

4.3 Decentralized Communication Sliding

In this section, we introduce a primal-dual algorithmic framework, namely, the decentralized communication sliding (DCS) method, for solving the saddle point problem (4.2.1) in a decentralized fashion. Moreover, we will establish complexity bounds on the required number of inter-node communication rounds as well as the total number of required subgradient evaluations. Throughout this section, we consider the deterministic case where exact subgradients of f_i's are available.

4.3.1 The DCS Algorithm

The basic scheme of the DCS algorithm is inspired by Chambolle and Pock's primal-dual method in [93]. The primal-dual method in [93] is an efficient and simple method for solving saddle point problems, which can be viewed as a refined version of the primal-dual hybrid gradient method by Arrow et al. [105]. However, its analysis is more closely related to a few recent important works for solving bilinear saddle point problems (e.g., [97, 106, 107, 108]). When applied to our saddle point reformulation defined in (4.2.1), for any given initial points $\mathbf{x}^0 = \mathbf{x}^{-1} \in X^m$ and $\mathbf{y}^0 \in \mathbb{R}^{md}$, and certain nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$, the primal-dual method updates $(\mathbf{x}^k, \mathbf{y}^k)$ according to

$$\tilde{\mathbf{x}}^k = \alpha_k(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}) + \mathbf{x}^{k-1}, \tag{4.3.1}$$

$$\mathbf{y}^k = \mathrm{argmin}_{\mathbf{y}\in\mathbb{R}^{md}} \langle -\mathbf{L}\tilde{\mathbf{x}}^k, \mathbf{y}\rangle + \tfrac{\tau_k}{2}\|\mathbf{y}-\mathbf{y}^{k-1}\|^2, \tag{4.3.2}$$

$$\mathbf{x}^k = \mathrm{argmin}_{\mathbf{x}\in X^m} \left\{\Phi^k(\mathbf{x}) := \langle \mathbf{L}\mathbf{y}^k, \mathbf{x}\rangle + F(\mathbf{x}) + \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x})\right\}. \tag{4.3.3}$$

Note that the incorporation of the Bregman distance into the primal-dual method (see (4.3.3)) was first introduced in [109].

In each iteration of the primal-dual method, only the computation of the matrix-vector products $\mathbf{L}\tilde{\mathbf{x}}^k$ and $\mathbf{L}\mathbf{y}^k$ will involve the communication among different agents, while the other computations such as the updating of $\tilde{\mathbf{x}}^k, \mathbf{y}^k$ and $\mathbf{x}^k$ can be performed separately by each agent. Under the assumption that the subproblem (4.3.3) can be easily solved, we can show that by properly choosing the algorithmic parameters α_k, τ_k and η_k one can find an ϵ-solution, i.e., a point $\bar{\mathbf{x}} \in X^m$ such that $F(\bar{\mathbf{x}}) - F(\mathbf{x}^*) \leq \epsilon$ and $\|\mathbf{L}\bar{\mathbf{x}}\| \leq \epsilon$, within $\mathcal{O}(1/\epsilon)$ iterations ([97, 106, 107, 108, 26, 94, 110]). This implies that one can find such an ϵ-solution in $\mathcal{O}(1/\epsilon)$ rounds of communication, which already improves the existing $\mathcal{O}(1/\epsilon^2)$ communication complexity for decentralized nonsmooth optimization. However, such a communication complexity bound is not quite meaningful because F is a general nonsmooth convex function and it is often difficult to solve the primal subproblem (4.3.3) explicitly.

One natural way to address this issue is to approximately solve (4.3.3) through an iterative subgradient descent method. Inside this iterative subgradient descent method, we do not need to re-compute the matrix-vector products $\mathbf{L}\tilde{\mathbf{x}}^k$ and $\mathbf{L}\mathbf{y}^k$, and hence no communication cost is involved. However, a straightforward pursuit of this approach, i.e., to solve the subproblem accurately enough at each iteration, does not necessarily yield the best complexity bound in terms of the total number of subgradient computations. To achieve the best possible complexity bounds in terms of both subgradient computation and communication, the proposed DCS method (along with its analysis) are in fact more complicated than the aforementioned inexact primal-dual method in the following two aspects. Firstly, while in most inexact first-order methods one usually computes only one approximate solution of

the subproblems, in the proposed DCS method we need to generate a pair of closely related approximate solutions $\mathbf{x}^k = (x_1^k, \ldots, x_m^k)$ and $\hat{\mathbf{x}}^k = (\hat{x}_1^k, \ldots, \hat{x}_m^k)$ to the subproblem in (4.3.3). Secondly, we need to modify the primal-dual method in a way such that one of these sequence (i.e.,$\{\hat{\mathbf{x}}^k\}$) will be used in the the extrapolation step in (4.3.1), while the other sequence $\{\mathbf{x}^k\}$ will act as the prox-center in $\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x})$ (see (4.3.3)).

Algorithm 8 DCS from agent i's perspective

Let $x_i^0 = x_i^{-1} = \hat{x}_i^0 \in X_i$, $y_i^0 \in \mathbb{R}^d$ for $i \in [m]$ and the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{T_k\}$ be given.
for $k = 1, \ldots, N$ **do**
Update $z_i^k = (\hat{x}_i^k, y_i^k)$ according to

$$\tilde{x}_i^k = \alpha_k(\hat{x}_i^{k-1} - x_i^{k-2}) + x_i^{k-1}, \tag{4.3.4}$$
$$v_i^k = \textstyle\sum_{j \in N_i} \mathcal{L}_{ij} \tilde{x}_j^k, \tag{4.3.5}$$
$$y_i^k = \operatorname{argmin}_{y_i \in \mathbb{R}^d} \langle -v_i^k, y_i \rangle + \tfrac{\tau_k}{2}\|y_i - y_i^{k-1}\|^2 = y_i^{k-1} + \tfrac{1}{\tau_k} v_i^k, \tag{4.3.6}$$
$$w_i^k = \textstyle\sum_{j \in N_i} \mathcal{L}_{ij} y_j^k, \tag{4.3.7}$$
$$(x_i^k, \hat{x}_i^k) = \mathrm{CS}(f_i, X_i, V_i, T_k, \eta_k, w_i^k, x_i^{k-1}). \tag{4.3.8}$$

end for
Return $z_i^N = \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta z_i^k$

The CS (Communication-Sliding) procedure called at (4.3.8) is stated as follows.
procedure: $(x, \hat{x}) = \mathrm{CS}(\phi, U, V, T, \eta, w, x)$
Let $u^0 = \hat{u}^0 = x$ and the parameters $\{\beta_t\}$ and $\{\lambda_t\}$ be given.
for $t = 1, \ldots, T$ **do**

$$h^{t-1} = \phi'(u^{t-1}) \in \partial\phi(u^{t-1}), \tag{4.3.9}$$
$$u^t = \operatorname{argmin}_{u \in U} \left[\langle w + h^{t-1}, u\rangle + \eta V(x, u) + \eta\beta_t V(u^{t-1}, u)\right]. \tag{4.3.10}$$

end for
Set

$$\hat{u}^T := \left(\textstyle\sum_{t=1}^T \lambda_t\right)^{-1} \sum_{t=1}^T \lambda_t u^t. \tag{4.3.11}$$

Set $x = u^T$ and $\hat{x} = \hat{u}^T$.
end procedure

We formally describe our DCS method in Algorithm 8. An outer iteration of the DCS algorithm occurs whenever the index k in Algorithm 8 is incremented by 1. More specif-

ically, each primal estimate x_i^0 is locally initialized from some arbitrary point in X_i, and x_i^{-1} and $\hat{x}_i^0$ are also set to be the same value. At each time step $k \geq 1$, each agent $i \in \mathcal{N}$ computes a local prediction $\tilde{x}_i^k$ using these three previous primal iterates (ref. (4.3.4)), and sends it to all of the nodes in its neighborhood, i.e., to all agents $j \in N_i$. In (4.3.5)-(4.3.6), each agent i then calculates the neighborhood disagreement v_i^k using the messages received from agents in N_i, and updates the dual subvector y_i^k. Then, another round of communication occurs in (4.3.7) when calculating w_i^k based on these updated dual variables. Therefore, each outer iteration k involves two communication rounds, one for the primal estimates and the other for the dual variables. Lastly, each agent i approximately solves the proximal projection subproblem (4.3.3), i.e.,

$$\text{argmin}_{u \in U} \, \langle w, u \rangle + \phi(u) + \eta V(x, u) \tag{4.3.12}$$

with $u = x_i$, $U = X_i$, $w = w_i^k$, $\phi = f_i$, $\eta = \eta_k$ and $V = V_i$, by calling the CS procedure for $T = T_k$ iterations in (4.3.8).

Each iteration performed by the CS procedure, referred to as an inner iteration of the DCS method, is equivalent to a subgradient descent step applied to (4.3.12). More specifically, each inner iteration consists of the computation of the subgradient $\phi'(u^{t-1})$ in (4.3.9) and the solution of the projection subproblem in (4.3.10). Note that the objective function of (4.3.10) consists of two parts: 1) the inner product of u and the summation of w and the current subgradient $\phi'(u^{t-1})$; and 2) two Bregman distances requiring that the new iterate lies near x and u^{t-1}. By using the definition of Bregman distance, we can see that (4.3.10) is equivalent to

$$u^t = \text{argmin}_{u \in U} \left[\langle w + h^{t-1} - \eta \nabla \omega(x) - \eta \beta_t \nabla \omega(u^{t-1}), u \rangle + \eta(1 + \beta_t) \omega(u) \right].$$

Similar to mirror-descent type methods, we assume that this problem is easy to solve. Also observe that the same dual information $w = w_i^k$ (see (4.3.7)) has been used throughout the

$T = T_k$ iterations of the CS procedure, and hence no additional communication is required within the procedure, which explains the name of the DCS method.

Observe that the DCS method, in spirit, has been inspired by our recent work on gradient sliding [111]. However, the gradient sliding method in [111] focuses on how to save gradient evaluations for solving certain structured convex optimization problems, rather than how to save communication rounds (or matrix-vector products) for decentralized optimization, and its algorithmic scheme is also quite different from the DCS method. It should also be note that the description of the algorithm is only conceptual at this moment since we have not specified the parameters $\{\alpha_k\}$, $\{\eta_k\}$, $\{\tau_k\}$, $\{T_k\}$, $\{\beta_t\}$ and $\{\lambda_t\}$ yet. We will later instantiate this generic algorithm when we state its convergence properties.

4.3.2 Convergence of DCS on General Convex Functions

We now establish the main convergence properties of the DCS algorithm. More specifically, we provide in Lemma 4.3.2 an estimate on the gap function defined in (4.2.2) together with stepsize policies which work for the general nonsmooth convex case with $\mu = 0$ (cf. (1.2.14)). The proof of this lemma can be found in Section 4.5.

Lemma 4.3.2 *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1, \ldots, N$ be generated by Algorithm 8 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. If the objective f_i, $i = 1, \ldots, m$, are general nonsmooth convex functions, i.e., $\mu = 0$ and $M > 0$, let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 8 satisfy*

$$\theta_k \frac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \leq \theta_{k-1} \frac{(T_{k-1}+1)(T_{k-1}+2)\eta_{k-1}}{T_{k-1}(T_{k-1}+3)}, \quad k = 2, \ldots, N, \tag{4.3.13}$$

$$\alpha_k \theta_k = \theta_{k-1}, \quad k = 2, \ldots, N, \tag{4.3.14}$$

$$\theta_k \tau_k = \theta_1 \tau_1, \quad k = 2, \ldots, N, \tag{4.3.15}$$

$$\alpha_k \|\mathbf{L}\|^2 \leq \eta_{k-1}\tau_k, \quad k = 2, \ldots, N, \tag{4.3.16}$$

$$\theta_N \|\mathbf{L}\|^2 \leq \theta_1 \tau_1 \eta_N, \tag{4.3.17}$$

and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 8 be set to

$$\lambda_t = t+1, \quad \beta_t = \tfrac{t}{2}, \quad \forall t \geq 1. \tag{4.3.18}$$

Then, we have for all $\mathbf{z} := (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$,

$$Q(\hat{\mathbf{z}}^N; \mathbf{z}) \leq \left(\sum_{k=1}^N \theta_k\right)^{-1} \left[\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle + \sum_{k=1}^N \tfrac{4mM^2\theta_k}{(T_k+3)\eta_k}\right], \tag{4.3.19}$$

where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and Q is defined in (4.2.2)*. Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (4.2.1)*, we have*

$$\begin{aligned} &\tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right)\max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \\ &\qquad \leq \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 + \sum_{k=1}^N \tfrac{4mM^2\theta_k}{\eta_k(T_k+3)}. \end{aligned} \tag{4.3.20}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (4.3.13)-(4.3.17). Using Lemma 4.3.2 and Proposition 4.2.1, we also establish the complexity of the DCS method for computing an (ϵ, δ)-solution of problem (1.2.19) when the objective functions are general convex.

Theorem 4.3.3 *Let $\mathbf{x}^*$ be an optimal solution of (1.2.19), the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 8 be set to* (4.3.18)*, and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\alpha_k = \theta_k = 1,\ \eta_k = 2\|\mathbf{L}\|,\ \tau_k = \|\mathbf{L}\|,\ \text{and } T_k = \left\lceil \tfrac{mM^2N}{\|\mathbf{L}\|^2\tilde{D}}\right\rceil, \quad \forall k = 1, \ldots, N, \tag{4.3.21}$$

for some $\tilde{D} > 0$. Then, for any $N \geq 1$, we have

$$F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \leq \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 2\tilde{D}\right] \tag{4.3.22}$$

and

$$\|\mathbf{L}\hat{\mathbf{x}}^N\| \leq \frac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+4\tilde{D}}+4\|\mathbf{y}^*-\mathbf{y}^0\|\right], \tag{4.3.23}$$

where $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^N \hat{\mathbf{x}}^k$*, and* y^* *is an arbitrary dual optimal solution.*

Proof. It is easy to check that (4.3.21) satisfies conditions (4.3.13)-(4.3.17). Particularly,

$$\tfrac{(T_1+1)(T_1+2)}{T_1(T_1+3)} = 1+\tfrac{2}{T_1^2+3T_1} \leq \tfrac{3}{2}.$$

Therefore, by plugging in these values to (4.3.19), we have

$$Q(\hat{\mathbf{z}}^N;\mathbf{x}^*,\mathbf{y}) \leq \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{1}{2}\|\mathbf{y}^0\|^2+2\tilde{D}\right]+\tfrac{1}{N}\langle\hat{\mathbf{s}},\mathbf{y}\rangle. \tag{4.3.24}$$

Letting $\hat{\mathbf{s}}^N = \frac{1}{N}\hat{\mathbf{s}}$, then from (4.3.20), we have

$$\begin{aligned}\|\hat{\mathbf{s}}^N\| &\leq \tfrac{\|\mathbf{L}\|}{N}\left[\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|+\|\mathbf{y}^N-\mathbf{y}^*\|+\|\mathbf{y}^*-\mathbf{y}^0\|\right]\\ &\leq \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\|\mathbf{y}^*-\mathbf{y}^0\|^2+4\tilde{D}}+\|\mathbf{y}^*-\mathbf{y}^0\|\right].\end{aligned}$$

Furthermore, by (4.3.24), we have

$$g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N) \leq \tfrac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{1}{2}\|\mathbf{y}^0\|^2+2\tilde{D}\right].$$

Applying Proposition 4.2.1 to the above two inequalities, the results in (4.3.22) and (4.3.23) follow immediately. ∎

We now make some remarks about the results obtained in Theorem 4.3.3. Firstly, even though one can choose any $\tilde{D} > 0$ (e.g., $\tilde{D} = 1$) in (4.3.21), the best selection of $\tilde{D}$ would be $\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)$ so that the first and third terms in (4.3.24) are about the same order. In

practice, if there exists an estimate $\mathcal{D}_{X^m} > 0$ s.t.

$$\mathbf{V}(\mathbf{x}_1, \mathbf{x}_2) \leq \mathcal{D}^2_{X^m},\ \forall \mathbf{x}_1, \mathbf{x}_2 \in X^m, \tag{4.3.25}$$

then we can set $\tilde{D} = \mathcal{D}^2_{X^m}$.

Secondly, the complexity of the DCS method directly follows from (4.3.22) and (4.3.23). For simplicity, let us assume that X is bounded, $\tilde{D} = \mathcal{D}^2_{X^m}$ and $\mathbf{y}^0 = \mathbf{0}$. We can see that the total number of inter-node communication rounds and intra-node subgradient evaluations required by each agent for finding an (ϵ, δ)-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\|\mathbf{L}\| \max\left(\tfrac{\mathcal{D}^2_{X^m}}{\epsilon}, \tfrac{\mathcal{D}_{X^m}+\|\mathbf{y}^*\|}{\delta}\right)\right\} \text{ and } \mathcal{O}\left\{mM^2 \max\left(\tfrac{\mathcal{D}^2_{X^m}}{\epsilon^2}, \tfrac{\mathcal{D}^2_{X^m}+\|\mathbf{y}^*\|^2}{\mathcal{D}^2_{X^m}\delta^2}\right)\right\}, \tag{4.3.26}$$

respectively. In particular, if ϵ and δ satisfy

$$\tfrac{\epsilon}{\delta} \leq \tfrac{\mathcal{D}^2_{X^m}}{\mathcal{D}_{X^m}+\|\mathbf{y}^*\|}, \tag{4.3.27}$$

then the previous two complexity bounds in (4.3.26), respectively, reduce to

$$\mathcal{O}\left\{\tfrac{\|\mathbf{L}\|\mathcal{D}^2_{X^m}}{\epsilon}\right\} \text{ and } \mathcal{O}\left\{\tfrac{mM^2\mathcal{D}^2_{X^m}}{\epsilon^2}\right\}. \tag{4.3.28}$$

Thirdly, it is interesting to compare DCS with the centralized mirror descent method [23] applied to (1.2.13). In the worst case, the Lipschitz constant of f in (1.2.13) can be bounded by $M_f \leq mM$, and each iteration of the method will incur m subgradient evaluations. Hence, the total number of subgradient evaluations performed by the mirror descent method for finding an ϵ-solution of (1.2.13), i.e., a point $\bar{x} \in X$ such that $f(\bar{x}) - f^* \leq \epsilon$, can be bounded by

$$\mathcal{O}\left\{\tfrac{m^3M^2\mathcal{D}^2_X}{\epsilon^2}\right\}, \tag{4.3.29}$$

where $\mathcal{D}^2_X$ characterizes the diameter of X, i.e., $\mathcal{D}^2_X := \max_{x_1,x_2\in X} V(x_1, x_2)$. Noting

that $\mathcal{D}_X^2/\mathcal{D}_{X^m}^2 = \mathcal{O}(1/m)$, and that the second bound in (4.3.28) states only the number of subgradient evaluations for each agent in the DCS method, we conclude that the total number of subgradient evaluations performed by DCS is comparable to the classic mirror descent method as long as (4.3.27) holds and hence not improvable in general.

Finally, observe that the parameter setting (4.3.21) requires the knowledge of the norm of Laplacian matrix $\mathbf{L}$, i.e., $\|\mathbf{L}\| = \max_{\|x\|\leq 1}\{\|\mathbf{L}\mathbf{x}\|_2\}$. If we use l_2-norm for the primal space, $\|\mathbf{L}\|$ will be the maximum eigenvalue of L. We can estimate it using power iteration method or simply bound it by the maximum degree of the graph. If we use l_1-norm in the primal space, then $\|\mathbf{L}\|$ will be the $L_{1,2}$-norm for $\|\mathbf{L}\|$, i.e., $\|\mathbf{L}\| = \|\mathbf{L}\|_{1,2} = \left(\sum_{i=1}^{md}\|\mathbf{L}_i\|_1^2\right)^{1/2} = 2\sqrt{d\sum_{j=1}^{m}\text{deg}_j^2}$, where $\mathbf{L}_i$'s denote the row vectors of $\mathbf{L}$ and deg_j denotes the degree of node j. The estimation of $\|\mathbf{L}\|$ will involve a few rounds of communication, however, these initial setup costs are independent of the target accuracy ϵ of the solution. It should also be noted that the number of inner iterations T_k given in (4.3.21) is fixed as a constant in order to achieve the best complexity bounds. In practice, it is reasonable to choose T_k dynamically so that a smaller number of inner iterations will be performed in the first few outer iterations. One simple strategy would be to set

$$T_k = \min\left(ck, \left\lceil \frac{mM^2N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil\right)$$

for some constant $c > 0$. While theoretically such a selection of T_k will result in slightly worse complexity bounds (up to an $\mathcal{O}(\log(1/\epsilon))$ factor) in terms of subgradient computations and communication rounds, it may improve the practical performance of the DCS method especially in the beginning of the execution of this method.

4.3.3 Boundedness of $\|\mathbf{y}^*\|$

In this subsection, we will provide a bound on the optimal dual multiplier $\mathbf{y}^*$. By doing so, we show that the complexity of DCS algorithm (as well as the stochastic DCS algorithm

in Section 4.4) only depends on the parameters for the primal problem along with the smallest nonzero eigenvalue of $\mathbf{L}$ and the initial point $\mathbf{y}^0$, even though these algorithms are intrinsically primal-dual type methods.

Theorem 4.3.4 *Let* $\mathbf{x}^*$ *be an optimal solution of* (1.2.19). *Then there exists an optimal dual multiplier* $\mathbf{y}^*$ *for* (4.2.1) *s.t.*

$$\|\mathbf{y}^*\| \leq \frac{\sqrt{m}M}{\tilde{\sigma}_{min}(\mathbf{L})}, \tag{4.3.30}$$

where $\tilde{\sigma}_{min}(\mathbf{L})$ *denotes the smallest nonzero eigenvalue of* $\mathbf{L}$.

Proof. Since we only relax the linear constraints in problem (1.2.19) to obtain the Lagrange dual problem (4.2.1), it follows from the strong Lagrange duality and the existence of $\mathbf{x}^*$ to (1.2.19) that an optimal dual multiplier $\mathbf{y}^*$ for problem (4.2.1) must exist. It is clear that

$$\mathbf{y}^* = \mathbf{y}_N^* + \mathbf{y}_C^*,$$

where $\mathbf{y}_N^*$ and $\mathbf{y}_C^*$ denote the projections of $\mathbf{y}^*$ over the null space and the column space of $\mathbf{L}^T$, respectively.

We consider two cases. Case 1) $\mathbf{y}_C^* = \mathbf{0}$. Since $\mathbf{y}_N^*$ belongs to the null space of $\mathbf{L}^T$, $\mathbf{L}^T\mathbf{y}^* = \mathbf{L}^T\mathbf{y}_N^* = \mathbf{0}$, which implies that for any $c \in \mathbb{R}$, $c\mathbf{y}^*$ is also an optimal dual multiplier of (4.2.1). Therefore, (4.3.30) clearly holds, because we can scale $\mathbf{y}^*$ to an arbitrary small vector.

Case 2) $\mathbf{y}_C^* \neq \mathbf{0}$. Using the fact that $\mathbf{L}^T\mathbf{y}^* = \mathbf{L}^T\mathbf{y}_C^*$ and the definition of a saddle point of (4.2.1), we conclude that $\mathbf{y}_C^*$ is also an optimal dual multiplier of (4.2.1). Since $\mathbf{y}_C^*$ in the column space of $\mathbf{L}$, we have

$$\|\mathbf{L}^T\mathbf{y}_C^*\|^2 = (\mathbf{y}_C^*)^T\mathbf{L}\mathbf{L}^T\mathbf{y}_C^* = (\mathbf{y}_C^*)^T\mathbf{U}^T\mathbf{\Lambda}\mathbf{U}\mathbf{y}_C^* \geq \tilde{\lambda}_{\min}(\mathbf{L}\mathbf{L}^T)\|\mathbf{U}\mathbf{y}_C^*\|^2 = \tilde{\sigma}_{\min}^2(\mathbf{L})\|\mathbf{y}_C^*\|^2,$$

where $\mathbf{U}$ is an orthonormal matrix whose rows consist of the eigenvectors of $\mathbf{L}\mathbf{L}^T$, $\mathbf{\Lambda}$ is the

diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\tilde{\lambda}_{min}(\mathbf{L}\mathbf{L}^T)$ denotes the smallest nonzero eigenvalue of $\mathbf{L}\mathbf{L}^T$, and $\tilde{\sigma}_{min}(\mathbf{L})$ denotes the smallest nonzero eigenvalue of $\mathbf{L}$. In particular,

$$\|\mathbf{y}_C^*\| \leq \tfrac{\|\mathbf{L}^T\mathbf{y}_C^*\|}{\tilde{\sigma}_{\min}(\mathbf{L})}. \tag{4.3.31}$$

Moreover, if we denote the saddle point problem defined in (4.2.1) as follows:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) := F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}, \mathbf{y} \rangle.$$

By the definition of a saddle point of (4.2.1), we have $\mathcal{L}(\mathbf{x}^*, \mathbf{y}_C^*) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}_C^*)$, i.e.,

$$F(\mathbf{x}^*) - F(\mathbf{x}) \leq \langle -\mathbf{L}^T\mathbf{y}_C^*, \mathbf{x} - \mathbf{x}^* \rangle.$$

Hence, from the definition of subgradients, we conclude that $-\mathbf{L}^T\mathbf{y}_C^* \in \partial F(\mathbf{x}^*)$, which together with the fact that $F(\cdot)$ is Lipschitz continuous implies that

$$\|\mathbf{L}^T\mathbf{y}_C^*\| = \|\textstyle\sum_{i=1}^m f_i'(x_i^*)\| \leq \sqrt{m}M.$$

Our result in (4.3.30) follows immediately from the above relation, (4.3.31) and the fact that $\mathbf{y}_C^*$ is also an optimal dual multiplier of (4.2.1). ∎

Observe that our bound for the dual multiplier $\mathbf{y}^*$ in (4.3.30) contains only the primal information. Given an initial dual multiplier $\mathbf{y}^0$, this result can be used to provide an upper bound on $\|\mathbf{y}^0 - \mathbf{y}^*\|$ in Theorems 4.3.3-4.4.10 throughout this chapter. Note also that we can assume $\mathbf{y}^0 = 0$ to simplify these complexity bounds.

4.3.4 Convergence of DCS on Strongly Convex Functions

In this subsection, we assume that the objective functions f_i's are strongly convex (i.e., $\mu > 0$ (1.2.14)). In order to take advantage of the strong convexity of the objective functions, we assume that the prox-functions $V_i(\cdot, \cdot)$, $i = 1, \ldots, m$, (cf. (4.2.7)) are growing quadratically

with the *quadratic growth constant* $\mathcal{C}$, i.e., there exists a constant $\mathcal{C} > 0$ such that

$$V_i(x_i, u_i) \leq \tfrac{\mathcal{C}}{2}\|x_i - u_i\|_{X_i}^2, \quad \forall x_i, u_i \in X_i,\ i = 1, \ldots, m. \tag{4.3.32}$$

By (4.2.8), we must have $\mathcal{C} \geq 1$.

We next provide in Lemma 4.3.5 an estimate on the gap function defined in (4.2.2) together with stepsize policies which work for the strongly convex case. The proof of this lemma can be found in Section 4.5.

Lemma 4.3.5 *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1, \ldots, N$ be generated by Algorithm 8 and $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$. If the objective f_i, $i = 1, \ldots, m$ are strongly convex functions, i.e., $\mu, M > 0$, let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$ and $\{\tau_k\}$ in Algorithm 8 satisfy* (4.3.14)-(4.3.17) *and*

$$\theta_k\eta_k \leq \theta_{k-1}(\mu/\mathcal{C} + \eta_{k-1}), \quad k = 2, \ldots, N, \tag{4.3.33}$$

and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 8 be set to

$$\lambda_t = t, \quad \beta_t^{(k)} = \tfrac{(t+1)\mu}{2\eta_k\mathcal{C}} + \tfrac{t-1}{2}, \quad \forall t \geq 1. \tag{4.3.34}$$

Then, we have for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$

$$\begin{aligned} Q(\hat{\mathbf{z}}^N; \mathbf{z}) \leq \left(\textstyle\sum_{k=1}^N \theta_k\right)^{-1} \Bigg[& \theta_1\eta_1\mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \\ & + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \frac{2mM^2\theta_k}{T_k(T_k+1)} \frac{t}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \Bigg], \end{aligned} \tag{4.3.35}$$

where $\hat{\mathbf{s}} := \theta_N\mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and Q is defined in (4.2.2)*. Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (4.2.1)*, we have*

$$\tfrac{\theta_N}{2}\left(1 - \tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \tag{4.3.36}$$

$$\leq \theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 + \textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\frac{2mM^2\theta_k}{T_k(T_k+1)}\frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}.$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (4.3.14)-(4.3.17) and (4.3.33). Also, by using Lemma 4.3.5 and Proposition 4.2.1, we establish the complexity of the DCS method for computing an (ϵ,δ)-solution of problem (1.2.19) when the objective functions are strongly convex. The choice of variable stepsizes rather than using constant stepsizes will accelerate its convergence rate.

Theorem 4.3.6 *Let $\mathbf{x}^*$ be an optimal solution of (1.2.19), the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 8 be set to* (4.3.34) *and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to*

$$\alpha_k = \tfrac{k}{k+1},\ \theta_k = k+1,\ \eta_k = \tfrac{k\mu}{2\mathcal{C}},\ \tau_k = \tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{(k+1)\mu},\ \textit{and } T_k = \left\lceil \sqrt{\tfrac{2m}{\tilde{D}}}\tfrac{\mathcal{C}MN}{\mu}\max\left\{\sqrt{\tfrac{2m}{\tilde{D}}}\tfrac{4\mathcal{C}M}{\mu}, 1\right\}\right\rceil, \tag{4.3.37}$$

$\forall k = 1,\ldots,N$, for some $\tilde{D} > 0$. Then, for any $N \geq 2$, we have

$$F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \leq \tfrac{2}{N(N+3)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2 + \tfrac{2\mu\tilde{D}}{\mathcal{C}}\right], \tag{4.3.38}$$

and

$$\|\mathbf{L}\hat{\mathbf{x}}^N\| \leq \tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)} + \tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\|\right], \tag{4.3.39}$$

where $\hat{\mathbf{x}}^N = \frac{2}{N(N+3)}\sum_{k=1}^N(k+1)\hat{\mathbf{x}}^k$, and y^* is an arbitrary dual optimal solution.

Proof. It is easy to check that (4.3.37) satisfies conditions (4.3.14)-(4.3.17) and (4.3.33). Moreover, we have

$$\begin{aligned}\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\frac{2mM^2\theta_k}{T_k(T_k+1)}\frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k} &= \textstyle\sum_{k=1}^N\frac{2mM^2\theta_k\mathcal{C}}{T_k(T_k+1)\mu}\sum_{t=1}^{T_k}\frac{2t}{2(t+1)+(t-1)k}\\ &\leq \textstyle\sum_{k=1}^N\frac{2mM^2\theta_k\mathcal{C}}{T_k(T_k+1)\mu}\left(\frac{1}{2} + \sum_{t=2}^{T_k}\frac{2t}{(t-1)(k+1)}\right)\end{aligned}$$

$$\leq \sum_{k=1}^{N} \frac{mM^2\mathcal{C}(k+1)}{T_k(T_k+1)\mu} + \sum_{k=1}^{N} \frac{8mM^2\mathcal{C}(T_k-1)}{T_k(T_k+1)\mu} \leq \frac{2\mu\tilde{D}}{\mathcal{C}}.$$

Therefore, by plugging in these values to (4.3.35), we have

$$Q(\hat{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y}) \leq \frac{2}{N(N+3)} \left[\frac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^0\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}} \right] + \frac{2}{N(N+3)} \langle \hat{\mathbf{s}}, \mathbf{y} \rangle. \quad (4.3.40)$$

Furthermore, from (4.3.36), we have for $N \geq 2$

$$\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2 \leq \frac{8\mathcal{C}}{\mu(N+1)(N-1)} \left[\frac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^0 - \mathbf{y}^*\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}} \right], \quad (4.3.41)$$

$$\|\mathbf{y}^* - \mathbf{y}^N\|^2 \leq \frac{N\mu}{(N-1)\|\mathbf{L}\|^2\mathcal{C}} \left[\frac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^0 - \mathbf{y}^*\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}} \right].$$

Let $\mathbf{s}^N := \frac{2}{N(N+3)}\hat{\mathbf{s}}$, then by using (4.3.41), we have for $N \geq 2$

$$\begin{aligned}
\|\mathbf{s}^N\| &\leq \frac{2}{N(N+3)} \left[(N+1)\|\mathbf{L}\| \|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \frac{4\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^N - \mathbf{y}^*\| + \frac{4\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^* - \mathbf{y}^0\| \right] \\
&\leq \frac{8\|\mathbf{L}\|}{N(N+3)} \left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2} \|\mathbf{y}^0 - \mathbf{y}^*\|^2} + \frac{\|\mathbf{L}\|\mathcal{C}}{\mu} \|\mathbf{y}^* - \mathbf{y}^0\| \right] \\
&\leq \frac{8\|\mathbf{L}\|}{N(N+3)} \left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)} + \frac{7\|\mathbf{L}\|\mathcal{C}}{\mu} \|\mathbf{y}^* - \mathbf{y}^0\| \right].
\end{aligned}$$

From (4.3.40), we further have

$$g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N) \leq \frac{2}{N(N+3)} \left[\frac{\mu}{\mathcal{C}} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu} \|\mathbf{y}^0\|^2 + \frac{2\mu\tilde{D}}{\mathcal{C}} \right].$$

Applying Proposition 4.2.1 to the above two inequalities, the results in (4.3.38) and (4.3.39) follow immediately. ∎

We now make some remarks about the results obtained in Theorem 4.3.6. Firstly, similar to the general convex case, the best choice for $\tilde{D}$ (cf. (4.3.37)) would be $\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$ so that the first and the third terms in (4.3.40) are about the same order. If there exists an estimate $\mathcal{D}_{X^m} > 0$ satisfying (4.3.25), we can set $\tilde{D} = \mathcal{D}_{X^m}^2$.

Secondly, the complexity of the DCS method for solving strongly convex problems fol-

lows from (4.3.38) and (4.3.39). For simplicity, let us assume that X is bounded, $\tilde{D} = \mathcal{D}_{X^m}^2$ and $\mathbf{y}^0 = \mathbf{0}$. We can see that the total number of inter-node communication rounds and intra-node subgradient evaluations performed by each agent for finding an (ϵ, δ)-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\max\left(\sqrt{\tfrac{\mu\mathcal{D}_{X^m}^2}{\mathcal{C}\epsilon}}, \sqrt{\tfrac{\|\mathbf{L}\|}{\delta}\left(\mathcal{D}_{X^m} + \tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mu}\right)}\right)\right\},$$

and

$$\mathcal{O}\left\{\tfrac{mM^2\mathcal{C}}{\mu}\max\left(\tfrac{1}{\epsilon}, \tfrac{\|\mathbf{L}\|\mathcal{C}}{\mu\delta}\left(\tfrac{1}{\mathcal{D}_{X^m}} + \tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mathcal{D}_{X^m}^2\mu}\right)\right)\right\}, \tag{4.3.42}$$

respectively. In particular, if ϵ and δ satisfy

$$\tfrac{\epsilon}{\delta} \leq \tfrac{\mu^2\mathcal{D}_{X^m}^2}{\|\mathbf{L}\|\mathcal{C}(\mu\mathcal{D}_{X^m} + \mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|)}, \tag{4.3.43}$$

then the complexity bounds in (4.3.42), respectively, reduce to

$$\mathcal{O}\left\{\sqrt{\tfrac{\mu\mathcal{D}_{X^m}^2}{\mathcal{C}\epsilon}}\right\} \text{ and } \mathcal{O}\left\{\tfrac{mM^2\mathcal{C}}{\mu\epsilon}\right\}. \tag{4.3.44}$$

Thirdly, we compare DCS method with the centralized mirror descent method [23] applied to (1.2.13). In the worst case, the Lipschitz constant and strongly convex modulus of f in (1.2.13) can be bounded by $M_f \leq mM$, and $\mu_f \geq m\mu$, respectively, and each iteration of the method will incur m subgradient evaluations. Therefore, the total number of subgradient evaluations performed by the mirror descent method for finding an ϵ-solution of (1.2.13), i.e., a point $\bar{x} \in X$ such that $f(\bar{x}) - f^* \leq \epsilon$, can be bounded by

$$\mathcal{O}\left\{\tfrac{m^2M^2\mathcal{C}}{\mu\epsilon}\right\}. \tag{4.3.45}$$

Observed that the second bound in (4.3.44) states only the number of subgradient evalua-

tions for each agent in the DCS method, we conclude that the total number of subgradient evaluations performed by DCS is comparable to the classic mirror descent method as long as (4.3.43) holds and hence not improvable in general for the nonsmooth strongly convex case.

4.4 Stochastic Decentralized Communication Sliding

In this section, we consider the stochastic case where only the noisy subgradient information of the functions f_i, $i = 1, \ldots, m$, is available or easier to compute. This situation happens when the function f_i's are given either in the form of expectation or as the summation of lots of components. This setting has attracted considerable interest in recent decades for its applications in a broad spectrum of disciplines including machine learning, signal processing, and operations research. We present a stochastic communication sliding method, namely the stochastic decentralized communication sliding (SDCS) method, and show that the similar complexity bounds as in Section 4.3 can still be obtained in expectation or with high probability.

4.4.1 The SDCS Algorithm

The first-order information of the function f_i, $i = 1, \ldots, m$, can be accessed by a stochastic first-order oracle ($\mathcal{SFO}$), which, given a point $u^t \in X$, outputs a vector $G_i(u^t, \xi_i^t)$ such that

$$\mathbb{E}[G_i(u^t, \xi_i^t)] = f_i'(u^t) \in \partial f_i(u^t), \tag{4.4.1}$$

$$\mathbb{E}[\|G_i(u^t, \xi_i^t) - f_i'(u^t)\|_*^2] \leq \sigma^2, \tag{4.4.2}$$

where ξ_i^t is a random vector which models a source of uncertainty and is independent of the search point u^t, and the distribution $\mathbb{P}(\xi_i)$ is not known in advance. We call $G_i(u^t, \xi_i^t)$ a *stochastic subgradient* of f_i at u^t.

The SDCS method can be obtained by simply replacing the exact subgradients in the

CS procedure of Algorithm 8 with the stochastic subgradients obtained from $\mathcal{SFO}$. This difference is described in Algorithm 9.

Algorithm 9 SDCS

The projection step (4.3.9)-(4.3.10) in the CS procedure of Algorithm 8 is replaced by

$$h^{t-1} = H(u^{t-1}, \xi^{t-1}), \tag{4.4.3}$$

$$u^t = \text{argmin}_{u \in U} \left[\langle w + h^{t-1}, u \rangle + \eta V(x, u) + \eta \beta_t V(u^{t-1}, u) \right], \tag{4.4.4}$$

where $H(u^{t-1}, \xi^{t-1})$ is a stochastic subgradient of ϕ at u^{t-1}.

We add a few remarks about the SDCS algorithm. Firstly, as in DCS, no additional communications of the dual variables are required when the subgradient projection (4.4.4) is performed for T_k times in the inner loop. This is because the same w_i^k has been used throughout the T_k iterations of the Stochastic CS procedure. Secondly, the problem will reduce to the deterministic case if there is no stochastic noise associated with the $\mathcal{SFO}$, i.e., when $\sigma = 0$ in (4.4.2). Therefore, in Section 4.5, we investigate the convergence analysis for the stochastic case first and then simplify the analysis for the deterministic case by setting $\sigma = 0$.

4.4.2 Convergence of SDCS on General Convex Functions

We now establish the main convergence properties of the SDCS algorithm. More specifically, we provide in Lemma 4.4.7 an estimate on the gap function defined in (4.2.2) together with stepsize policies which work for the general convex case with $\mu = 0$ (cf. (1.2.14)). The proof of this lemma can be found in Section 4.5.

Lemma 4.4.7 *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$ for $k = 1, \ldots, N$ be generated by Algorithm 9, $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k \right)^{-1} \sum_{k=1}^N \theta_k (\hat{\mathbf{x}}^k, \mathbf{y}^k)$, and Assumptions* (4.4.1)-(4.4.2) *hold. If the objective f_i, $i = 1, \ldots, m$, are general nonsmooth convex functions, i.e., $\mu = 0$ and $M > 0$, let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 9 satisfy* (4.3.13)-(4.3.17)*, and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 9 be*

set as (4.3.18)*. Then, for all* $\mathbf{z} \in X^m \times \mathbb{R}^{md}$*,*

$$Q(\hat{\mathbf{z}}^N; \mathbf{z}) \leq \left(\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \right. \tag{4.4.5}$$
$$\left. + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)} \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \frac{4(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{\eta_k} \right] \right\},$$

where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ *and* Q *is defined in* (4.2.2)*. Furthermore, for any saddle point* $(\mathbf{x}^*, \mathbf{y}^*)$ *of* (4.2.1)*, we have*

$$\frac{\theta_N}{2}\left(1 - \frac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right) \max\{\eta_N\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2, \tau_N\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \tag{4.4.6}$$
$$\leq \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)} \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2$$
$$+ \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)} \left[(t+1)\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \frac{4(M^2 + \|\delta_i^{t-1,k}\|_*^2)}{\eta_k} \right].$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (4.3.13)-(4.3.17). Also, by using Lemma 4.4.7 and Proposition 4.2.1, we establish the complexity of the SDCS method for computing an (ϵ, δ)-solution of problem (1.2.19) in expectation when the objective functions are general convex.

Theorem 4.4.8 *Let* $\mathbf{x}^*$ *be an optimal solution of (1.2.19), the parameters* $\{\lambda_t\}$ *and* $\{\beta_t\}$ *in the CS procedure of Algorithm 9 be set as* (4.3.18)*, and suppose that* $\{\alpha_k\}$*,* $\{\theta_k\}$*,* $\{\eta_k\}$*,* $\{\tau_k\}$ *and* $\{T_k\}$ *are set to*

$$\alpha_k = \theta_k = 1,\ \eta_k = 2\|\mathbf{L}\|,\ \tau_k = \|\mathbf{L}\|,\ \text{and } T_k = \left\lceil \frac{m(M^2+\sigma^2)N}{\|\mathbf{L}\|^2\tilde{D}} \right\rceil, \quad \forall k = 1, \ldots, N, \tag{4.4.7}$$

for some $\tilde{D} > 0$*. Then, under Assumptions* (4.4.1) *and* (4.4.2)*, we have for any* $N \geq 1$

$$\mathbb{E}[F(\hat{\mathbf{x}}^k) - F(\mathbf{x}^*)] \leq \frac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 + 4\tilde{D}\right], \tag{4.4.8}$$

and

$$\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}^N\|] \leq \frac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+8\tilde{D}}+4\|\mathbf{y}^*-\mathbf{y}^0\|\right]. \tag{4.4.9}$$

where $\hat{\mathbf{x}}^N = \frac{1}{N}\sum_{k=1}^N \hat{\mathbf{x}}^k$*, and* y^* *is an arbitrary dual optimal solution.*

Proof. It is easy to check that (4.4.7) satisfies conditions (4.3.13)-(4.3.17). Moreover, by (4.2.4), we can obtain

$$\begin{aligned} g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N) &= \max_{\mathbf{y}} Q(\hat{\mathbf{z}}^N;\mathbf{x}^*,\mathbf{y}) - \left(\sum_{k=1}^N\theta_k\right)^{-1}\langle\hat{\mathbf{s}},\mathbf{y}\rangle \qquad (4.4.10)\\ &\leq \left(\sum_{k=1}^N\theta_k\right)^{-1}\left\{\frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2\right.\\ &\quad\left.+\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right]\right\}, \end{aligned}$$

where $\mathbf{s}^N = \left(\sum_{k=1}^N\theta_k\right)^{-1}\hat{\mathbf{s}}$. Particularly, from Assumption (4.4.1) and (4.4.2),

$$\mathbb{E}[\delta_i^{t-1,k}]=0,\ \ \mathbb{E}[\|\delta_i^{t-1,k}\|_*^2]\leq\sigma^2,\ \forall i\in\{1,\ldots,m\},\ t\geq 1,\ k\geq 1,$$

and from (4.4.7)

$$\frac{(T_1+1)(T_1+2)}{T_1(T_1+3)} = 1+\frac{2}{T_1^2+3T_1}\leq\frac{3}{2}.$$

Therefore, by taking expectation over both sides of (4.4.10) and plugging in these values into (4.4.10), we have

$$\begin{aligned} \mathbb{E}[g(\hat{\mathbf{s}}^N,\hat{\mathbf{z}}^N)] &\leq \left(\sum_{k=1}^N\theta_k\right)^{-1}\left\{\frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\sum_{k=1}^N\frac{8m(M^2+\sigma^2)\theta_k}{(T_k+3)\eta_k}\right\} \qquad (4.4.11)\\ &\leq \frac{\|\mathbf{L}\|}{N}\left[3\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\frac{1}{2}\|\mathbf{y}^0\|^2+4\tilde{D}\right], \end{aligned}$$

with

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] = \tfrac{1}{N}\mathbb{E}[\|\hat{\mathbf{s}}\|] \leq \tfrac{\|\mathbf{L}\|}{N}\mathbb{E}\left[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

Note that from (4.4.6) and Jensen's inequality, we have

$$(\mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|])^2 \leq \mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2] \leq 6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^* - \mathbf{y}^0\| + 8\tilde{D},$$
$$(\mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|])^2 \leq \mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|^2] \leq 12\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 2\|\mathbf{y}^* - \mathbf{y}^0\| + 16\tilde{D}.$$

Hence,

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] \leq \tfrac{\|\mathbf{L}\|}{N}\left[3\sqrt{6\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + 8\tilde{D}} + 4\|\mathbf{y}^* - \mathbf{y}^0\|\right].$$

Applying Proposition 4.2.1 to the above inequality and (4.4.11), the results in (4.4.8) and (4.4.9) follow immediately. ∎

We now make some observations about the results obtained in Theorem 4.4.8. Firstly, one can choose any $\tilde{D} > 0$ (e.g., $\tilde{D} = 1$) in (4.4.7), however, the best selection of $\tilde{D}$ would be $\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$ so that the first and third terms in (4.4.11) are about the same order. In practice, if there exists an estimate $\mathcal{D}_{X^m} > 0$ satisfying (4.3.25), we can set $\tilde{D} = \mathcal{D}_{X^m}^2$.

Secondly, the complexity of SDCS method immediately follows from (4.4.8) and (4.4.9). Under the above assumption, with $\tilde{D} = \mathcal{D}_{X^m}^2$ and $\mathbf{y}^0 = \mathbf{0}$, we can see that the total number of inter-node communication rounds and intra-node subgradient evaluations required by each agent for finding a stochastic (ϵ, δ)-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\|\mathbf{L}\|\max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon}, \tfrac{\mathcal{D}_{X^m}+\|\mathbf{y}^*\|}{\delta}\right)\right\} \text{ and } \mathcal{O}\left\{m(M^2+\sigma^2)\max\left(\tfrac{\mathcal{D}_{X^m}^2}{\epsilon^2}, \tfrac{\mathcal{D}_{X^m}^2+\|\mathbf{y}^*\|^2}{\mathcal{D}_{X^m}^2\delta^2}\right)\right\}, \tag{4.4.12}$$

respectively. In particular, if ϵ and δ satisfy (4.3.27), the above complexity bounds, respec-

tively, reduce to

$$\mathcal{O}\left\{\frac{\|\mathbf{L}\|\mathcal{D}_{Xm}^2}{\epsilon}\right\} \text{ and } \mathcal{O}\left\{\frac{m(M^2+\sigma^2)\mathcal{D}_{Xm}^2}{\epsilon^2}\right\}. \tag{4.4.13}$$

In particular, we can show that the total number stochastic subgradients that SDCS requires is comparable to the mirror-descent stochastic approximation in [3]. This implies that the sample complexity for decentralized stochastic optimization are still optimal (as the centralized one), even after we skip many communication rounds.

4.4.3 Convergence of SDCS on Strongly Convex Functions

We now provide in Lemma 4.4.9 an estimate on the gap function defined in (4.2.2) together with stepsize policies which work for the strongly convex case with $\mu > 0$ (cf. (1.2.14)). The proof of this lemma can be found in Section 4.5.

Note that throughout this subsection, we assume that the prox-functions $V_i(\cdot,\cdot)$, $i = 1,\ldots,m$, (cf. (4.2.7)) are growing quadratically with the quadratic growth constant $\mathcal{C}$, i.e., (4.3.32) holds.

Lemma 4.4.9 *Let the iterates $(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, $k = 1,\ldots,N$ be generated by Algorithm 9, $\hat{\mathbf{z}}^N$ be defined as $\hat{\mathbf{z}}^N := \left(\sum_{k=1}^N \theta_k\right)^{-1} \sum_{k=1}^N \theta_k(\hat{\mathbf{x}}^k, \mathbf{y}^k)$, and Assumptions* (4.4.1)-(4.4.2) *hold. If the objective f_i, $i = 1,\ldots,m$ are strongly convex functions, i.e., $\mu, M > 0$, let the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$ and $\{\tau_k\}$ in Algorithm 9 satisfy* (4.3.14)-(4.3.17) *and* (4.3.33)*, and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 9 be set as* (4.3.34)*. Then, for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,*

$$\begin{aligned} Q(\hat{\mathbf{z}}^N;\mathbf{z}) \;\le\; &\left(\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0,\mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \right. \\ &\left. + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \tfrac{2\theta_k}{T_k(T_k+1)} \left[t\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \tfrac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k} \right] \right\}, \end{aligned} \tag{4.4.14}$$

where $\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0)$ and Q is defined in (4.2.2)*. Furthermore,*

for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ *of* (4.2.1)*, we have*

$$\begin{aligned}\tfrac{\theta_N}{2}\left(1-\tfrac{\|\mathbf{L}\|^2}{\eta_N\tau_N}\right)\max\{\eta_N\|\hat{\mathbf{x}}^N-\mathbf{x}^{N-1}\|^2,\tau_N\|\mathbf{y}^*-\mathbf{y}^N\|^2\} \\ \leq \theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2 \\ +\textstyle\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\tfrac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\tfrac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right].\end{aligned} \tag{4.4.15}$$

In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ satisfying (4.3.14)-(4.3.17) and (4.3.13). Also, by using Lemma 4.4.9 and Proposition 4.2.1, we establish the complexity of the SDCS method for computing an (ϵ,δ)-solution of problem (1.2.19) in expectation when the objective functions are strongly convex. Similar to the deterministic case, we choose variable stepsizes rather than constant stepsizes.

Theorem 4.4.10 *Let* $\mathbf{x}^*$ *be an optimal solution of (1.2.19), the parameters* $\{\lambda_t\}$ *and* $\{\beta_t\}$ *in the CS procedure of Algorithm 9 be set as* (4.3.34)*, and suppose that* $\{\alpha_k\}$*,* $\{\theta_k\}$*,* $\{\eta_k\}$*,* $\{\tau_k\}$ *and* $\{T_k\}$ *are set to*

$$\begin{aligned}&\alpha_k=\tfrac{k}{k+1},\ \theta_k=k+1,\ \eta_k=\tfrac{k\mu}{2\mathcal{C}},\ \tau_k=\tfrac{4\|\mathbf{L}\|^2\mathcal{C}}{(k+1)\mu},\ \textit{and} \\ &T_k=\left\lceil\sqrt{\tfrac{m(M^2+\sigma^2)}{\tilde{D}}}\tfrac{2N\mathcal{C}}{\mu}\max\left\{\sqrt{\tfrac{m(M^2+\sigma^2)}{\tilde{D}}}\tfrac{8\mathcal{C}}{\mu},1\right\}\right\rceil,\quad\forall k=1,\ldots,N,\end{aligned} \tag{4.4.16}$$

for some $\tilde{D}>0$*. Then, under Assumptions* (4.4.1) *and* (4.4.2)*, we have for any* $N\geq 2$

$$\mathbb{E}[F(\bar{\mathbf{x}}^N)-F(\mathbf{x}^*)\leq\tfrac{2}{N(N+3)}\left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2+\tfrac{2\mu\tilde{D}}{\mathcal{C}}\right], \tag{4.4.17}$$

and

$$\mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}^N\|]\leq\tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D}+\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)}+\tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^*-\mathbf{y}^0\|\right], \tag{4.4.18}$$

where $\hat{\mathbf{x}}^N=\frac{2}{N(N+3)}\sum_{k=1}^N(k+1)\hat{\mathbf{x}}^k$*, and* y^* *is an arbitrary dual optimal solution.*

Proof. It is easy to check that (4.4.16) satisfies conditions (4.3.14)-(4.3.17) and (4.3.33).

Similarly, by (4.2.4), Assumption (4.4.1) and (4.4.2), we can obtain

$$\mathbb{E}[g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)] \leq \left(\sum_{k=1}^N \theta_k\right)^{-1} \Bigg\{ \theta_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \tfrac{2\theta_k}{T_k(T_k+1)} \left[\tfrac{2t(M^2+\sigma^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right] \Bigg\}, \tag{4.4.19}$$

where $\mathbf{s}^N = \left(\sum_{k=1}^N \theta_k\right)^{-1} \hat{\mathbf{s}}$. Particularly, from (4.4.16), we have

$$\begin{aligned}
\sum_{k=1}^N \sum_{t=1}^{T_k} \tfrac{4m(M^2+\sigma^2)\theta_k}{T_k(T_k+1)} \tfrac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k} &= \sum_{k=1}^N \tfrac{4m(M^2+\sigma^2)\mathcal{C}\theta_k}{T_k(T_k+1)\mu} \sum_{t=1}^{T_k} \tfrac{2t}{2(t+1)+(t-1)k} \\
&\leq \sum_{k=1}^N \tfrac{4m(M^2+\sigma^2)\mathcal{C}\theta_k}{T_k(T_k+1)\mu} \left(\tfrac{1}{2} + \sum_{t=2}^{T_k} \tfrac{2t}{(t-1)(k+1)}\right) \\
&\leq \sum_{k=1}^N \tfrac{2m(M^2+\sigma^2)\mathcal{C}(k+1)}{T_k(T_k+1)\mu} + \sum_{k=1}^N \tfrac{16m(M^2+\sigma^2)\mathcal{C}(T_k-1)}{T_k(T_k+1)\mu} \leq \tfrac{2\mu\tilde{D}}{\mathcal{C}}.
\end{aligned}$$

Therefore, by plugging in these values into (4.4.19), we have

$$\mathbb{E}[g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)] \leq \tfrac{2}{N(N+3)} \left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|L\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0\|^2 + \tfrac{2\mu\tilde{D}}{\mathcal{C}}\right], \tag{4.4.20}$$

with

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] = \tfrac{2}{N(N+3)}\mathbb{E}[\|\hat{\mathbf{s}}\|] \leq \tfrac{2\|\mathbf{L}\|}{N(N+3)}\mathbb{E}\left[(N+1)\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\| + \tfrac{4\|\mathbf{L}\|\mathcal{C}}{\mu}(\|\mathbf{y}^N - \mathbf{y}^*\| + \|\mathbf{y}^* - \mathbf{y}^0\|)\right].$$

Note that from (4.4.15), we have, for any $N \geq 2$,

$$\begin{aligned}
\mathbb{E}[\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2] &\leq \tfrac{8}{(N+1)(N-1)} \left[\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2}\|\mathbf{y}^0 - \mathbf{y}^*\|^2 + 2\tilde{D}\right], \\
\mathbb{E}[\|\mathbf{y}^* - \mathbf{y}^N\|^2] &\leq \tfrac{N\mu}{(N-1)\|\mathbf{L}\|^2\mathcal{C}} \left[\tfrac{\mu}{\mathcal{C}}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}}{\mu}\|\mathbf{y}^0 - \mathbf{y}^*\|^2 + \tfrac{2\mu\tilde{D}}{\mathcal{C}}\right].
\end{aligned}$$

Hence, in view of the above three relations and Jensen's inequality, we obtain

$$\mathbb{E}[\|\hat{\mathbf{s}}^N\|] \leq \tfrac{8\|\mathbf{L}\|}{N(N+3)} \left[3\sqrt{2\tilde{D} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\|\mathbf{L}\|^2\mathcal{C}^2}{\mu^2}\|\mathbf{y}^0 - \mathbf{y}^*\|^2} + \tfrac{\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^* - \mathbf{y}^0\|\right]$$

$$\leq \tfrac{8\|\mathbf{L}\|}{N(N+3)}\left[3\sqrt{2\tilde{D}+\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)}+\tfrac{7\|\mathbf{L}\|\mathcal{C}}{\mu}\|\mathbf{y}^*-\mathbf{y}^0\|\right].$$

Applying Proposition 4.2.1 to the above inequality and (4.4.20), the results in (4.4.17) and (4.4.18) follow immediately. ∎

We now make some observations about the results obtained in Theorem 4.4.10. Firstly, similar to the general convex case, the best choice for $\tilde{D}$ (cf. (4.4.16)) would be $\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)$ so that the first and the third terms in (4.4.20) are about the same order. If there exists an estimate $\mathcal{D}_{X^m}>0$ satisfying (4.3.25), we can set $\tilde{D}=\mathcal{D}^2_{X^m}$.

Secondly, the complexity of SDCS method for solving strongly convex problems follows from (4.4.17) and (4.4.18). Under the above assumption, with $\tilde{D}=\mathcal{D}^2_{X^m}$ and $\mathbf{y}^0=\mathbf{0}$, the total number of inter-node communication rounds and intra-node subgradient evaluations performed by each agent for finding a stochastic (ϵ,δ)-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\max\left(\sqrt{\tfrac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}},\sqrt{\tfrac{\|\mathbf{L}\|}{\delta}\left(\mathcal{D}_{X^m}+\tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mu}\right)}\right)\right\},$$

and

$$\mathcal{O}\left\{\tfrac{m(M^2+\sigma^2)\mathcal{C}}{\mu}\max\left(\tfrac{1}{\epsilon},\tfrac{\mathcal{C}\|\mathbf{L}\|}{\mu\delta}\left(\tfrac{1}{\mathcal{D}_{X^m}}+\tfrac{\mathcal{C}\|\mathbf{L}\|\|\mathbf{y}^*\|}{\mathcal{D}^2_{X^m}\mu}\right)\right)\right\}, \tag{4.4.21}$$

respectively. In particular, if ϵ and δ satisfy (4.3.43), the above complexity bounds, respectively, reduce to

$$\mathcal{O}\left\{\sqrt{\tfrac{\mu\mathcal{D}^2_{X^m}}{\mathcal{C}\epsilon}}\right\} \text{ and } \mathcal{O}\left\{\tfrac{m(M^2+\sigma^2)\mathcal{C}}{\mu\epsilon}\right\}. \tag{4.4.22}$$

We can see that the total number of stochastic subgradient computations is comparable to the optimal complexity bound obtained in [112, 12] for stochastic strongly convex case in the centralized case.

4.4.4 High Probability Results

All of the results stated in Section 4.4.2-4.4.3 are established in terms of expectation. In order to provide high probability results for SDCS method, we additionally need the following "light-tail" assumption:

$$\mathbb{E}[\exp\{\|G_i(u^t, \xi_i^t) - f_i'(u^t)\|_*^2/\sigma^2\}] \leq \exp\{1\}. \tag{4.4.23}$$

Note that (4.4.23) is stronger than (4.4.2), since it implies (4.4.2) by Jensen's inequality. Moreover, we also assume that there exists $\bar{\mathbf{V}}(\mathbf{x}^*)$ s.t.

$$\bar{\mathbf{V}}(\mathbf{x}^*) := \sum_{i=1}^m \bar{V}_i(x_i^*) := \sum_{i=1}^m \max_{x_i \in X_i} V_i(x_i^*, x_i). \tag{4.4.24}$$

The following theorem provides a large deviation result for the gap function $g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N)$ when our objective functions f_i, $i = 1, \ldots, m$ are general nonsmooth convex functions.

Theorem 4.4.11 *Let x^* be an optimal solution of* (1.2.19)*, Assumptions* (4.4.1)*,* (4.4.2) *and* (4.4.23) *hold, the parameters $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ in Algorithm 9 satisfy* (4.3.13)-(4.3.17)*, and the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 9 be set as* (4.3.18)*. In addition, if X_i's are compact, then for any $\zeta > 0$ and $N \geq 1$, we have*

$$\text{Prob}\left\{g(\hat{\mathbf{s}}^N, \hat{\mathbf{z}}^N) \geq \mathcal{B}_d(N) + \zeta \mathcal{B}_p(N)\right\} \leq \exp\{-\zeta^2/3\} + \exp\{-\zeta\}, \tag{4.4.25}$$

where

$$\mathcal{B}_d(N) := \left(\sum_{k=1}^N \theta_k\right)^{-1} \left[\frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \sum_{k=1}^N \frac{8m(M^2+\sigma^2)\theta_k}{\eta_k(T_k+3)}\right], \tag{4.4.26}$$

and

$$\mathcal{B}_p(N) := \left(\sum_{k=1}^N \theta_k\right)^{-1} \left\{ \sigma \left[2\bar{\mathbf{V}}(\mathbf{x}^*) \sum_{k=1}^N \sum_{t=1}^{T_k} \left(\frac{\theta_k \lambda_t}{\sum_{t=1}^{T_k} \lambda_t} \right)^2 \right]^{1/2} + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{\sigma^2 \theta_k \lambda_t}{\left(\sum_{t=1}^{T_k} \lambda_t\right) \eta_k \beta_t} \right\}. \tag{4.4.27}$$

In the next corollary, we establish the rate of convergence of SDCS in terms of both primal and feasibility (or consistency) residuals are of order $\mathcal{O}(1/N)$ with high probability when the objective functions are nonsmooth and convex.

Corollary 4.4.12 *Let $\mathbf{x}^*$ be an optimal solution of (1.2.19), y^* be an arbitrary dual optimal solution, the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the CS procedure of Algorithm 9 be set as* (4.3.18)*, and suppose that $\{\alpha_k\}$, $\{\theta_k\}$, $\{\eta_k\}$, $\{\tau_k\}$ and $\{T_k\}$ are set to* (4.4.7) *with $\tilde{D} = \bar{\mathbf{V}}(\mathbf{x}^*)$. Under Assumptions* (4.4.1)*,* (4.4.2) *and* (4.4.23)*, we have for any $N \geq 1$ and $\zeta > 0$*

$$\text{Prob}\left\{ F(\hat{\mathbf{x}}^N) - F(\mathbf{x}^*) \geq \tfrac{\|\mathbf{L}\|}{N} \left[(7 + 8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*) + \tfrac{1}{2}\|\mathbf{y}^0\|^2 \right] \right\} \leq \exp\{-\zeta^2/3\} + \exp\{-\zeta\}, \tag{4.4.28}$$

and

$$\text{Prob}\left\{ \|\mathbf{L}\hat{\mathbf{x}}^N\|^2 \geq \tfrac{18\|\mathbf{L}\|^2}{N^2} \left[(7 + 8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*) + \tfrac{2}{3}\|\mathbf{y}^* - \mathbf{y}^0\|^2 \right] \right\} \leq \exp\{-\zeta^2/3\} + \exp\{-\zeta\}. \tag{4.4.29}$$

Proof. Observe that by the definition of λ_t in (4.3.18),

$$\begin{aligned} \sum_{t=1}^{T_k} \left[\frac{\theta_k \lambda_t}{\sum_{t=1}^{T_k} \lambda_t} \right]^2 &= \left(\tfrac{2}{T_k(T_k+3)} \right)^2 \sum_{t=1}^{T_k} (t+1)^2 \\ &= \left(\tfrac{2}{T_k(T_k+3)} \right)^2 \tfrac{(T_k+1)(T_k+2)(2T_k+3)}{6} \leq \tfrac{8}{3T_k}, \end{aligned}$$

which together with (4.4.27) then imply that

$$\begin{aligned}\mathcal{B}_p(N) &\leq \tfrac{1}{N}\left\{\sigma\left[2\bar{\mathbf{V}}(\mathbf{x}^*)\textstyle\sum_{k=1}^{N}\frac{8}{3T_k}\right]^{1/2} + \sum_{k=1}^{N}\frac{8m\sigma^2}{\|\mathbf{L}\|(T_k+3)}\right\} \\ &\leq \tfrac{4\|\mathbf{L}\|}{N}\left\{\sqrt{\tfrac{\bar{\mathbf{V}}(\mathbf{x}^*)\tilde{D}}{3m}} + \tilde{D}\right\} \leq \tfrac{8\|\mathbf{L}\|\bar{\mathbf{V}}(\mathbf{x}^*)}{N}.\end{aligned}$$

Hence, (4.4.28) follows from the above relation, (4.4.25) and Proposition 4.2.1. Note that from (4.4.6) and plugging in (4.4.7) with $\tilde{D} = \bar{\mathbf{V}}(\mathbf{x}^*)$, we obtain

$$\begin{aligned}\|\hat{\mathbf{s}}^N\|^2 &= \left(\textstyle\sum_{k=1}^{N}\theta_k\right)^{-2}\|\hat{\mathbf{s}}\|^2 \\ &\leq \left(\textstyle\sum_{k=1}^{N}\theta_k\right)^{-2}\left\{3\theta_N^2\|\mathbf{L}\|^2\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2 + 3\theta_1^2\tau_1^2\left(\|\mathbf{y}^N - \mathbf{y}^*\|^2 + \|\mathbf{y}^* - \mathbf{y}^0\|^2\right)\right\} \\ &\leq \tfrac{3\|\mathbf{L}\|^2}{N^2}\Bigg\{18\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + 4\|\mathbf{y}^* - \mathbf{y}^0\|^2 \\ &\quad + \textstyle\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{12\theta_k}{T_k(T_k+3)\|\mathbf{L}\|}\left[(t+1)\langle\delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right]\Bigg\}.\end{aligned}$$

Hence, similarly, we have

$$\text{Prob}\left\{\|\hat{\mathbf{s}}^N\|^2 \geq \tfrac{18\|\mathbf{L}\|^2}{N^2}\left[(7+8\zeta)\bar{\mathbf{V}}(\mathbf{x}^*) + \tfrac{2}{3}\|\mathbf{y}^* - \mathbf{y}^0\|^2\right]\right\} \leq \exp\{-\zeta^2/3\} + \exp\{-\zeta\},$$

which in view of Proposition 4.2.1 immediately implies (4.4.29). ■

4.5 Convergence Analysis

This section is devoted to the proof of the main lemmas in Section 4.3 and 4.4, which establish the convergence results of the deterministic and stochastic decentralized communication sliding methods, respectively. After introducing some general results about these algorithms, we provide the proofs for Lemma 4.3.2-4.4.9 and Theorem 4.4.11.

Before we provide proofs for Lemma 4.3.2-4.4.9, we first need to present a result which summarizes an important convergence property of the CS procedure. It needs to be men-

tioned that the following proposition states a general result holds for CS procedure performed by individual agent $i \in \mathcal{N}$. For notation convenience, we use the notations defined in CS procedure (cf. Algorithm 8).

Proposition 4.5.13 *If* $\{\beta_t\}$ *and* $\{\lambda_t\}$ *in the* CS *procedure satisfy*

$$\lambda_{t+1}(\eta\beta_{t+1} - \mu/\mathcal{C}) \leq \lambda_t(1+\beta_t)\eta, \ \forall t \geq 1. \tag{4.5.1}$$

then, for $t \geq 1$ *and* $u \in U$,

$$\begin{aligned} &(\textstyle\sum_{t=1}^T \lambda_t)^{-1}\left[\eta(1+\beta_T)\lambda_T V(u^T,u) + \sum_{t=1}^T \lambda_t\langle\delta^{t-1}, u-u^{t-1}\rangle\right] + \Phi(\hat{u}^T) - \Phi(u) \\ &\leq (\textstyle\sum_{t=1}^T \lambda_t)^{-1}\left[(\eta\beta_1 - \mu/\mathcal{C})\lambda_1 V(u^0,u) + \sum_{t=1}^T \frac{(M+\|\delta^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}\right], \end{aligned} \tag{4.5.2}$$

where Φ *is defined as*

$$\Phi(u) := \langle w, u\rangle + \phi(u) + \eta V(x,u) \tag{4.5.3}$$

and $\delta^t := \phi'(u^t) - h^t$.

Proof. Noticing that $\phi := f_i$ in the CS procedure, we have by (1.2.14)

$$\begin{aligned} \phi(u^t) &\leq \phi(u^{t-1}) + \langle\phi'(u^{t-1}), u^t - u^{t-1}\rangle + M\|u^t - u^{t-1}\| \\ &= \phi(u^{t-1}) + \langle\phi'(u^{t-1}), u - u^{t-1}\rangle + \langle\phi'(u^{t-1}), u^t - u\rangle + M\|u^t - u^{t-1}\| \\ &\leq \phi(u) - \tfrac{\mu}{2}\|u - u^{t-1}\|^2 + \langle\phi'(u^{t-1}), u^t - u\rangle + M\|u^t - u^{t-1}\|, \end{aligned}$$

where $\phi'(u^{t-1}) \in \partial\phi(u^{t-1})$ and $\partial\phi(u^{t-1})$ denotes the subdifferential of ϕ at u^{t-1}. By applying Lemma A.0.1 to (4.3.10), we obtain

$$\langle w + h^{t-1}, u^t - u\rangle + \eta V(x,u^t) - \eta V(x,u)$$

$$\leq \eta\beta_t V(u^{t-1}, u) - \eta(1+\beta_t)V(u^t, u) - \eta\beta_t V(u^{t-1}, u^t),\ \forall u \in U.$$

Combining the above two relations together with (4.3.32) [2], we conclude that

$$\langle w, u^t - u\rangle + \phi(u^t) - \phi(u) + \langle \delta^{t-1}, u - u^{t-1}\rangle + \eta V(x, u^t) - \eta V(x, u) \tag{4.5.4}$$

$$\leq (\eta\beta_t - \mu/\mathcal{C})V(u^{t-1}, u) - \eta(1+\beta_t)V(u^t, u) + \langle \delta^{t-1}, u^t - u^{t-1}\rangle$$

$$+ M\|u^t - u^{t-1}\| - \eta\beta_t V(u^{t-1}, u^t),\ \forall u \in U. \tag{4.5.5}$$

Moreover, by Cauchy-Schwarz inequality, (4.2.8), and the simple fact that $-at^2/2 + bt \leq b^2/(2a)$ for any $a > 0$, we have

$$\langle \delta^{t-1}, u^t - u^{t-1}\rangle + M\|u^t - u^{t-1}\| - \eta\beta_t V(u^{t-1}, u^t) \leq (\|\delta^{t-1}\|_* + M)\|u^t - u^{t-1}\| - \tfrac{\eta\beta_t}{2}\|u^t - t^{t-1}\|^2$$
$$\leq \tfrac{(M+\|\delta^{t-1}\|_*)^2}{2\eta\beta_t}.$$

From the above relation and the definition of $\Phi(u)$ in (4.5.3), we can rewrite (4.5.4) as,

$$\Phi(u^t) - \Phi(u) + \langle \delta^{t-1}, u - u^{t-1}\rangle \leq (\eta\beta_t - \mu/\mathcal{C})V(u^{t-1}, u) - \eta(1+\beta_t)V(u^t, u) + \tfrac{(M+\|\delta^{t-1}\|_*)^2}{2\eta\beta_t}.$$

Multiplying both sides by λ_t and summing up the resulting inequalities from $t = 1$ to T, we obtain

$$\textstyle\sum_{t=1}^T \lambda_t \left[\Phi(u^t) - \Phi(u) + \langle \delta^{t-1}, u - u^{t-1}\rangle\right] \leq \sum_{t=1}^T \left[(\eta\beta_t - \mu/\mathcal{C})\lambda_t V(u^{t-1}, u) - \eta(1+\beta_t)\lambda_t V(u^t, u)\right]$$
$$+ \textstyle\sum_{t=1}^T \frac{(M+\|\delta^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}.$$

Hence, in view of (4.5.1), the convexity of Φ and the definition of $\hat{u}^T$ in (4.3.11), we have

$$\Phi(\hat{u}^T) - \Phi(u) + (\textstyle\sum_{t=1}^T \lambda_t)^{-1} \sum_{t=1}^T \lambda_t \langle \delta^{t-1}, u - u^{t-1}\rangle$$

[2] Observed that we only need condition (4.3.32) when $\mu > 0$, in other words, the objective functions f_i's are strongly convex.

$$\leq (\textstyle\sum_{t=1}^{T}\lambda_t)^{-1}\left[(\eta\beta_1-\mu/\mathcal{C})\lambda_1 V(u^0,u)-\eta(1+\beta_T)\lambda_T V(u^T,u)\right.$$
$$\left.+\textstyle\sum_{t=1}^{T}\frac{(M+\|\delta^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}\right],$$

which implies (4.5.2) immediately. ∎

As a matter of fact, the SDCS method covers the DCS method as a special case when $\delta^t = 0,\ \forall t \geq 0$. Therefore, we investigate the proofs for Lemma 4.4.7 and 4.4.9 first and then simplify them for the proofs for Lemma 4.3.2 and 4.3.5. We now provide a proof for Lemma 4.4.7, which establishes the convergence property of SDCS method for solving general convex problems.

Proof of Lemma 4.4.7

When f_i, $i = 1,\ldots,m$, are general convex functions, we have $\mu = 0$ and $M > 0$ (cf. (1.2.14)). Therefore, in view of $\phi := f_i$, and λ_t and β_t defined in (4.3.18) satisfying condition (4.5.1) in the CS procedure, equation (4.5.2) can be rewritten as the following,[3]

$$(\textstyle\sum_{t=1}^{T}\lambda_t)^{-1}\left[\eta(1+\beta_T)\lambda_T V_i(u_i^T,u_i)+\sum_{t=1}^{T}\lambda_t\langle\delta_i^{t-1},u_i-u_i^{t-1}\rangle\right]+\Phi_i(\hat{u}_i^T)-\Phi_i(u_i)$$
$$\leq (\textstyle\sum_{t=1}^{T}\lambda_t)^{-1}\left[\eta\beta_1\lambda_1 V_i(u_i^0,u_i)+\sum_{t=1}^{T}\frac{(M+\|\delta_i^{t-1}\|_*)^2\lambda_t}{2\eta\beta_t}\right],\ \forall u_i\in X_i.$$

In view of the above relation, the definition of Φ^k in (4.3.3), and the input and output settings in the CS procedure, it is not difficult to see that, for any $k \geq 1$,[4]

$$\Phi^k(\hat{\mathbf{x}}^k)-\Phi^k(\mathbf{x})\ +(\textstyle\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[\eta_k(1+\beta_{T_k})\lambda_{T_k}\mathbf{V}(\mathbf{x}^k,\mathbf{x})+\sum_{t=1}^{T_k}\sum_{i=1}^{m}\lambda_t\langle\delta_i^{t-1,k},x_i-u_i^{t-1}\rangle\right]$$
$$\leq (\textstyle\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[\eta_k\beta_1\lambda_1\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})+\sum_{t=1}^{T_k}\sum_{i=1}^{m}\frac{(M+\|\delta_i^{t-1,k}\|_*)^2\lambda_t}{2\eta_k\beta_t}\right],\ \forall\mathbf{x}\in X^m.$$

By plugging into the above relation the values of λ_t and β_t in (4.3.18), together with the

[3] We added the subscript i to emphasize that this inequality holds for any agent $i \in \mathcal{N}$ with $\phi = f_i$. More specifically, $\Phi_i(u_i) := \langle w_i, u_i\rangle + f_i(u_i) + \eta V_i(x_i,u_i)$.

[4] We added the superscript k in $\delta_i^{t-1,k}$ to emphasize that this error is generated at the k-th outer loop.

definition of Φ^k in (4.3.3) and rearranging the terms, we have,

$$\langle \mathbf{L}(\hat{\mathbf{x}}^k - \mathbf{x}), \mathbf{y}^k \rangle + F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) \leq \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\mathbf{x}^k, \mathbf{x})\right] - \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) + \tfrac{2}{T_k(T_k+3)} \sum_{t=1}^{T_k} \sum_{i=1}^{m} \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1} \rangle + \tfrac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right].$$

Moreover, applying Lemma A.0.1 to (4.3.6), we have, for $k \geq 1$,

$$\langle v_i^k, y_i - y_i^k \rangle \leq \tfrac{\tau_k}{2} \left[\|y_i - y_i^{k-1}\|^2 - \|y_i - y_i^k\|^2 - \|y_i^{k-1} - y_i^k\|^2\right], \ \forall y_i \in \mathbb{R}^d, \tag{4.5.6}$$

which in view of the definition of Q in (4.2.2) and the above two relations, then implies that, for $k \geq 1$, $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,

$$\begin{aligned} Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) &= F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\mathbf{x}}^k, \mathbf{y} \rangle - \langle \mathbf{L}\mathbf{x}, \mathbf{y}^k \rangle \\ &\leq \langle \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y} - \mathbf{y}^k \rangle + \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\mathbf{x}^k, \mathbf{x})\right] \\ &\quad - \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) + \tfrac{\tau_k}{2} \left[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \mathbf{y}^k\|^2 - \|\mathbf{y}^{k-1} - \mathbf{y}^k\|^2\right] \\ &\quad + \tfrac{2}{T_k(T_k+3)} \sum_{t=1}^{T_k} \sum_{i=1}^{m} \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1} \rangle + \tfrac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right]. \end{aligned}$$

Multiplying both sides of the above inequality by θ_k, and summing up the resulting inequalities from $k = 1$ to N, we obtain, for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,

$$\begin{aligned} \textstyle\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) &\leq \textstyle\sum_{k=1}^N \theta_k \Delta_k \\ &\quad + \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^{m} \tfrac{2\theta_k}{T_k(T_k+3)} \left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1} \rangle + \tfrac{2(M+\|\delta_i^{t-1,k}\|_*)^2}{\eta_k}\right], \end{aligned} \tag{4.5.7}$$

where

$$\begin{aligned} \Delta_k :=& \langle \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y} - \mathbf{y}^k \rangle + \tfrac{(T_k+1)(T_k+2)\eta_k}{T_k(T_k+3)} \left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\mathbf{x}^k, \mathbf{x})\right] \\ &- \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) + \tfrac{\tau_k}{2} \left[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \mathbf{y}^k\|^2 - \|\mathbf{y}^{k-1} - \mathbf{y}^k\|^2\right]. \end{aligned} \tag{4.5.8}$$

We now provide a bound on $\sum_{k=1}^{N}\theta_k\Delta_k$. Observe that from the definition of $\tilde{\mathbf{x}}^k$ in (4.3.1), (4.3.13) and (4.3.15) we have

$$\begin{aligned}
\textstyle\sum_{k=1}^{N}\theta_k\Delta_k &\leq \textstyle\sum_{k=1}^{N}\left[\theta_k\langle\mathbf{L}(\mathbf{x}^k-\mathbf{x}^{k-1}),\mathbf{y}-\mathbf{y}^k\rangle-\alpha_k\theta_k\langle\mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}-\mathbf{y}^{k-1}\rangle\right]\\
&\quad-\textstyle\sum_{k=1}^{N}\theta_k\left[\alpha_k\langle\mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^{k-1}-\mathbf{y}^k\rangle+\eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}^k)+\tfrac{\tau_k}{2}\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right]\\
&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N,\mathbf{x})\\
&\quad+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2 \qquad (4.5.9)\\
&\overset{(a)}{\leq}\theta_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
&\quad-\sum_{k=2}^{N}\left[\theta_k\alpha_k\langle\mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^{k-1}-\mathbf{y}^k\rangle+\theta_{k-1}\eta_{k-1}\mathbf{V}(\mathbf{x}^{k-2},\mathbf{x}^{k-1})+\tfrac{\theta_k\tau_k}{2}\|\mathbf{y}^{k-1}-\mathbf{y}^k\|^2\right]\\
&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N,\mathbf{x})\\
&\overset{(b)}{\leq}\theta_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
&\quad+\textstyle\sum_{k=2}^{N}\left(\frac{\theta_{k-1}\alpha_k\|\mathbf{L}\|^2}{2\tau_k}-\frac{\theta_{k-1}\eta_{k-1}}{2}\right)\|\mathbf{x}^{k-2}-\mathbf{x}^{k-1}\|^2\\
&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N,\mathbf{x})\\
&\overset{(c)}{\leq}\theta_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{\theta_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2\\
&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})-\tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N,\mathbf{x})\\
&\overset{(d)}{\leq}\theta_N\langle\mathbf{y}^N,\mathbf{L}(\mathbf{x}^{N-1}-\mathbf{x}^N)\rangle-\theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1},\mathbf{x}^N)-\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^N\|^2\\
&\quad+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\mathbf{y},\theta_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\theta_1\tau_1(\mathbf{y}^N-\mathbf{y}^0)\rangle,\\
&\overset{(e)}{\leq}\left(\tfrac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}-\tfrac{\theta_1\tau_1}{2}\right)\|\mathbf{y}^N\|^2+\tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2\\
&\quad+\langle\mathbf{y},\theta_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\theta_1\tau_1(\mathbf{y}^N-\mathbf{y}^0)\rangle, \qquad (4.5.10)
\end{aligned}$$

where (a) follows from (4.3.14) and the fact that $\mathbf{x}^{-1}=\mathbf{x}^0$, (b) follows from the simple relation that $b\langle u,v\rangle-a\|v\|^2/2\leq b^2\|u\|^2/(2a),\forall a>0$, (4.3.14) and (4.2.11), (c) follows from (4.3.16), (d) follows from (4.3.15), $\|\mathbf{y}-\mathbf{y}^0\|^2-\|\mathbf{y}-\mathbf{y}^N\|^2=\|\mathbf{y}^0\|^2-\|\mathbf{y}^N\|^2-2\langle\mathbf{y},\mathbf{y}^0-\mathbf{y}^N\rangle$ and arranging the terms accordingly, (e) follows from (4.2.11) and the re-

lation $b\langle u,v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a), \forall a > 0$. Using the above bound in (4.5.7) we obtain $\forall \mathbf{z} \in X^m \times \mathbb{R}^{md}$,

$$\begin{aligned}\textstyle\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) \leq{}& \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y}\rangle \\ &+ \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k}\sum_{i=1}^m \tfrac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],\end{aligned} \tag{4.5.11}$$

where

$$\hat{\mathbf{s}} := \theta_N \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}) + \theta_1\tau_1(\mathbf{y}^N - \mathbf{y}^0). \tag{4.5.12}$$

Our result in (4.4.5) immediately follows from the convexity of Q. Furthermore, in view of (4.5.9)(c) and (4.5.7), we can obtain the following result,

$$\begin{aligned}\textstyle\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) \leq{}& \theta_N\langle \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N\rangle - \theta_N\eta_N\mathbf{V}(\mathbf{x}^{N-1}, \hat{\mathbf{x}}^N) \\ &+ \tfrac{\theta_1\tau_1}{2}\|\mathbf{y} - \mathbf{y}^0\|^2 - \tfrac{\theta_N\tau_N}{2}\|\mathbf{y} - \mathbf{y}^N\|^2 \\ &+ \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}) - \tfrac{(T_N+1)(T_N+2)\theta_N\eta_N}{T_N(T_N+3)}\mathbf{V}(\mathbf{x}^N, \mathbf{x}) \\ &+ \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k}\sum_{i=1}^m \tfrac{\theta_k}{T_k(T_k+3)}\left[(t+1)\langle \delta_i^{t-1,k}, x_i - u_i^{t-1}\rangle + \tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right].\end{aligned}$$

Therefore, in view of the fact that $\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}^*) \geq 0$ for any saddle point $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ of (4.2.1), and (4.2.11), by fixing $\mathbf{z} = \mathbf{z}^*$ and rearranging terms, we obtain

$$\begin{aligned}\tfrac{\theta_N\eta_N}{2}\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2 \leq{}& \theta_N\langle \mathbf{L}(\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}), \mathbf{y}^* - \mathbf{y}^N\rangle - \tfrac{\theta_N\tau_N}{2}\|\mathbf{y}^* - \mathbf{y}^N\|^2 \\ &+ \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 \\ &+ \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k}\sum_{i=1}^m \tfrac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right] \\ \leq{}& \tfrac{\theta_N\|\mathbf{L}\|^2}{2\tau_N}\|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2 + \tfrac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1\tau_1}{2}\|\mathbf{y}^* - \mathbf{y}^0\|^2 \\ &+ \textstyle\sum_{k=1}^N \sum_{t=1}^{T_k}\sum_{i=1}^m \tfrac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1}\rangle + \tfrac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],\end{aligned} \tag{4.5.13}$$

where the second inequality follows from the relation $b\langle u,v\rangle - a\|v\|^2/2 \le b^2\|u\|^2/(2a), \forall a > 0$.

Similarly, we obtain

$$\frac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2 \le \frac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}\|\mathbf{y}^*-\mathbf{y}^N\|^2 + \frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \frac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2 \tag{4.5.14}$$
$$+ \sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+3)}\left[(t+1)\langle\delta_i^{t-1,k}, x_i^*-u_i^{t-1}\rangle + \frac{4(M^2+\|\delta_i^{t-1,k}\|_*^2)}{\eta_k}\right],$$

from which the desired result in (4.4.6) follows. ■

The following proof of Lemma 4.4.9 establishes the convergence of SDCS method for solving strongly convex problems.

Proof of Lemma 4.4.9

When f_i, $i = 1,\ldots,m$, are strongly convex functions, we have $\mu,\ M > 0$ (cf. (1.2.14)). Therefore, in view of Proposition 4.5.13 with λ_t and β_t defined in (4.3.34) satisfying condition (4.5.1), the definition of Φ^k in (4.3.3), and the input and output settings in the CS procedure, we have for all $k \ge 1, \forall \mathbf{x} \in X^m$

$$\Phi^k(\hat{\mathbf{x}}^k) - \Phi^k(\mathbf{x}) + (\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[\eta_k(1+\beta_{T_k}^{(k)})\lambda_{T_k}\mathbf{V}(\mathbf{x}^k,\mathbf{x}) + \sum_{t=1}^{T_k}\sum_{i=1}^m\lambda_t\langle\delta_i^{t-1,k}, x_i-u_i^{t-1}\rangle\right]$$
$$\le (\sum_{t=1}^{T_k}\lambda_t)^{-1}\left[(\eta_k\beta_1^{(k)}-\mu/\mathcal{C})\lambda_1\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) + \sum_{t=1}^{T_k}\sum_{i=1}^m\frac{(M+\|\delta_i^{t-1,k}\|_*)^2\lambda_t}{2\eta_k\beta_t}\right].$$

By plugging into the above relation the values of λ_t and $\beta_t^{(k)}$ in (4.3.34), together with the definition of Φ^k in (4.3.3) and rearranging the terms, we have $\forall \mathbf{x} \in X^m,\ k \ge 1$,

$$\langle\mathbf{L}(\hat{\mathbf{x}}^k-\mathbf{x}), \mathbf{y}^k\rangle + F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) \le \eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - (\mu/\mathcal{C}+\eta_k)\mathbf{V}(\mathbf{x}^k,\mathbf{x}) - \eta_k\mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)$$
$$+ \frac{2}{T_k(T_k+1)}\sum_{t=1}^{T_k}\sum_{i=1}^m\left[t\langle\delta_i^{t-1,k}, x_i-u_i^{t-1}\rangle + \frac{(M+\|\delta_i^{t-1,k}\|_*)^2 t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right].$$

In view of (4.5.6), the above relation and the definition of Q in (4.2.2), and following the

same trick that we used to obtain (4.5.7), we have, for all $\mathbf{z} \in X^m \times \mathbb{R}^{md}$,

$$\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) \leq \sum_{k=1}^N \theta_k \bar{\Delta}_k + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[t\langle \delta_i^{t-1,k}, x_i - u_i^{t-1} \rangle + \frac{(M+\|\delta_i^{t-1,k}\|_*)^2 t}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right], \tag{4.5.15}$$

where

$$\bar{\Delta}_k := \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y} - \mathbf{y}^k \rangle + \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - (\mu/\mathcal{C} + \eta_k)\mathbf{V}(\mathbf{x}^k, \mathbf{x}) - \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \hat{\mathbf{x}}^k) + \tfrac{\tau_k}{2} \left[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \mathbf{y}^k\|^2 - \|\mathbf{y}^{k-1} - \mathbf{y}^k\|^2 \right]. \tag{4.5.16}$$

Since $\bar{\Delta}_k$ in (4.5.16) shares a similar structure with Δ_k in (4.5.8), we can follow similar procedure as in (4.5.9) to simplify the RHS of (4.5.15). Note that the only difference of (4.5.16) and (4.5.8) is in the coefficient of the terms $\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x})$, and $\mathbf{V}(\mathbf{x}^k, \mathbf{x})$. Hence, by using condition (4.3.33) in place of (4.3.13), we obtain $\forall \mathbf{z} \in X^m \times \mathbb{R}^{md}$

$$\sum_{k=1}^N \theta_k Q(\hat{\mathbf{x}}^k, \mathbf{y}^k; \mathbf{z}) \leq \theta_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{\theta_1 \tau_1}{2} \|\mathbf{y}^0\|^2 + \langle \hat{\mathbf{s}}, \mathbf{y} \rangle + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[t\langle \delta_i^{t-1,k}, x_i - u_i^{t-1} \rangle + \frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right], \tag{4.5.17}$$

where $\hat{\mathbf{s}}$ is defined in (4.5.12). Our result in (4.4.14) immediately follows from the convexity of Q.

Following the same procedure as we obtain (4.5.13), for any saddle point $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ of (4.2.1), we have

$$\tfrac{\theta_N \eta_N}{2} \|\hat{\mathbf{x}}^N - \mathbf{x}^{N-1}\|^2 \leq \tfrac{\theta_N \|\mathbf{L}\|^2}{2\tau_N} \|\mathbf{x}^N - \mathbf{x}^{N-1}\|^2 + \theta_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{\theta_1 \tau_1}{2} \|\mathbf{y}^* - \mathbf{y}^0\|^2 + \sum_{k=1}^N \sum_{t=1}^{T_k} \sum_{i=1}^m \frac{2\theta_k}{T_k(T_k+1)} \left[t\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1} \rangle + \frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C} + (t-1)\eta_k} \right], \tag{4.5.18}$$

$$\frac{\theta_N\tau_N}{2}\|\mathbf{y}^*-\mathbf{y}^N\|^2 \le \frac{\theta_N\|\mathbf{L}\|^2}{2\eta_N}\|\mathbf{y}^*-\mathbf{y}^N\|^2+\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^*-\mathbf{y}^0\|^2$$
$$+\sum_{k=1}^N\sum_{t=1}^{T_k}\sum_{i=1}^m\frac{2\theta_k}{T_k(T_k+1)}\left[t\langle\delta_i^{t-1,k},x_i^*-u_i^{t-1}\rangle+\frac{2t(M^2+\|\delta_i^{t-1,k}\|_*^2)}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k}\right],$$

from which the desired result in (4.4.15) follows. ∎

We are ready to provide proofs for Lemma 4.3.2 and 4.3.5, which demonstrates the convergence properties of the deterministic communication sliding method.

Proof of Lemma 4.3.2

When $f_i,\ i=1,\ldots,m$ are general nonsmooth convex functions, we have $\delta_i^t=0$, $\mu=0$ and $M>0$. Therefore, in view of (4.5.11), we have, $\forall\mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\le\frac{(T_1+1)(T_1+2)\theta_1\eta_1}{T_1(T_1+3)}\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle+\sum_{k=1}^N\frac{4mM^2\theta_k}{(T_k+3)\eta_k},$$

where $\hat{\mathbf{s}}$ is defined in (4.5.12). Our result in (4.3.19) immediately follows from the convexity of Q. Moreover, our result in (4.3.20) follows from setting $\delta_i^{t-1,k}=0$ in (4.5.13) and (4.5.14). ∎

Proof of Lemma 4.3.5

When $f_i,\ i=1,\ldots,m$ are strongly convex functions, we have $\delta_i^t=0$ and $\mu,\ M>0$. Therefore, in view of (4.5.17), we obtain, $\forall\mathbf{z}\in X^m\times\mathbb{R}^{md}$,

$$\sum_{k=1}^N\theta_k Q(\hat{\mathbf{x}}^k,\mathbf{y}^k;\mathbf{z})\le\theta_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\frac{\theta_1\tau_1}{2}\|\mathbf{y}^0\|^2+\langle\hat{\mathbf{s}},\mathbf{y}\rangle+\sum_{k=1}^N\sum_{t=1}^{T_k}\frac{2mM^2\theta_k}{T_k(T_k+1)}\frac{t}{(t+1)\mu/\mathcal{C}+(t-1)\eta_k},$$

where $\hat{\mathbf{s}}$ is defined in (4.5.12). Our result in (4.3.35) immediately follows from the convexity of Q. Also, the result in (4.3.36) follows by setting $\delta_i^{t-1,k}=0$ in (4.5.18). ∎

We now provide a proof for Theorem 4.4.11 that establishes a large deviation result for the gap function.

Proof of Theorem 4.4.11:

Observe that by Assumption (4.4.1), (4.4.2) and (4.4.23) on the SO and the definition of $u_i^{t,k}$, the sequence $\{\langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1,k}\rangle\}_{1\le i\le m, 1\le t\le T_k, k\ge 1}$ is a martingale-difference sequence. Denoting

$$\gamma_{k,t} := \frac{\theta_k \lambda_t}{\sum_{t=1}^{T_k} \lambda_t},$$

and using the large-deviation theorem for martingale-difference sequence (e.g. Lemma 2 of [95]) and the fact that

$$\begin{aligned}
&\mathbb{E}[\exp\{\gamma_{k,t}^2 \langle \delta_i^{t-1,k}, x_i^* - u_i^{t-1,k}\rangle^2/(2\gamma_{k,t}^2 \bar{V}_i(x_i^*)\sigma^2)\}] \\
&\quad \le \mathbb{E}[\exp\{\|\delta_i^{t-1,k}\|_*^2, \|x_i^* - u_i^{t-1,k}\|^2/(2\bar{V}_i(x_i^*)\sigma^2)\}] \\
&\quad \le \mathbb{E}[\exp\{\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}] \le \exp\{1\},
\end{aligned}$$

we conclude that, $\forall \zeta > 0$,

$$\text{Prob}\left\{\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} \gamma_{k,t}\langle \delta_i^{t-1,k}, u_i^{t-1,k} - x_i^*\rangle > \zeta\sigma\sqrt{2\bar{\mathbf{V}}(\mathbf{x}^*)\sum_{k=1}^{N}\sum_{t=1}^{T_k}\gamma_{k,t}^2}\right\} \le \exp\{-\zeta^2/3\}. \tag{4.5.19}$$

Now let

$$S_{k,t} := \frac{\theta_k \lambda_t}{\left(\sum_{t=1}^{T_k}\lambda_t\right)\eta_k \beta_t},$$

and $S := \sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}$. By the convexity of exponential function, we have

$$\mathbb{E}[\exp\{\tfrac{1}{S}\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}] \le \mathbb{E}[\tfrac{1}{S}\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}\exp\{\|\delta_i^{t-1,k}\|_*^2/\sigma^2\}] \le \exp\{1\},$$

where the last inequality follows from Assumption (4.4.23). Therefore, by Markov's inequality, for all $\zeta > 0$,

$$\text{Prob}\left\{\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}\|\delta_i^{t-1,k}\|_*^2 > (1+\zeta)\sigma^2\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}\right\} \tag{4.5.20}$$

$$= \text{Prob}\left\{\exp\left\{\frac{1}{S}\sum_{k=1}^{N}\sum_{t=1}^{T_k}\sum_{i=1}^{m} S_{k,t}\|\delta_i^{t-1,k}\|_*^2/\sigma^2\right\} \geq \exp\{1+\zeta\}\right\} \leq \exp\{-\zeta\}.$$

Combing (4.5.19), (4.5.20), (4.4.5) and (4.2.4), our result in (4.4.25) immediately follows.

■

4.6 Numerical Results

In this section, we demonstrate the advantages of our (stochastic) decentralized communication sliding method over distributed dual averaging method proposed in [51] through some preliminary numerical experiments.

Let us consider the decentralized linear Support Vector Machines (SVM) model with the following hinge loss function

$$\max\{0, 1 - v\langle x, u\rangle\}, \tag{4.6.1}$$

where $(v, u) \in \mathbb{R} \times \mathbb{R}^d$ is the pair of class label and feature vector, and $x \in \mathbb{R}^d$ denotes the weight vector. Clearly, the hinge loss function is convex and nonsmooth with respect to x. For convex case, we study 1-norm SVM problem [113, 114], i.e., the hinge loss function (4.6.1) plus $l1$-norm as the regularizer, while for strongly convex case, we study 2-norm SVM model. Moreover, we use the Erhos-Renyi algorithm [5] to generate the underlying decentralized network, i.e., a connected graph with $m = 100$ nodes shown in Figure 4.1. Note that nodes with different degrees are drawn in different colors, in particular, nodes in red have maximum degree of 4. We also used the real dataset named "ijcnn1" from LIBSVM[6] and choose $20,000$ samples from this dataset as our problem instance data to train the decentralized SVM model. Since we have $m = 100$ nodes (or agents) in the decentralized network, we evenly split these $20,000$ samples over 100 nodes, and hence

[5] We implemented the Erhos-Renyi algorithm based on a MATLAB function written by Pablo Blider, which can be found in `https://www.mathworks.com/matlabcentral/fileexchange/4206`.

[6] This real dataset can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

each network node has 200 samples.

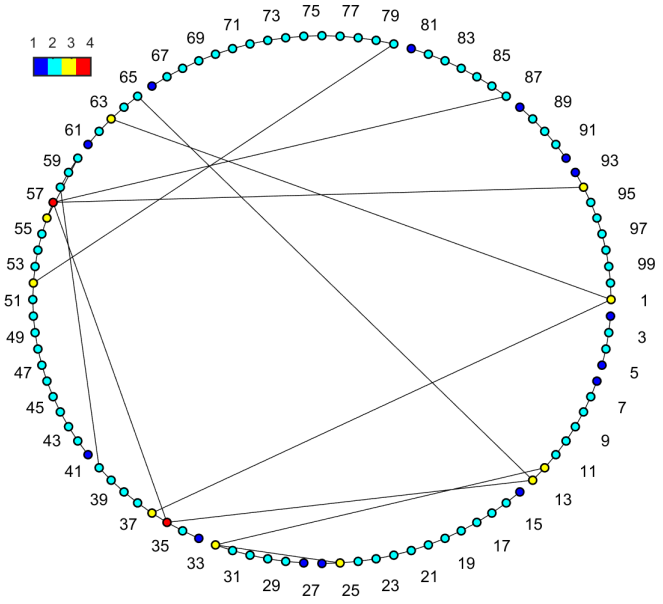

Figure 4.1: The underlying decentralized network

With the same initial points $x^0 = \mathbf{1}$ and $y^0 = \mathbf{0}$, we compare the performances of our algorithms with the distributed dual averaging method [51] for solving (1.2.13)-(1.2.19) by showing the progress of objective values versus the number of communication rounds and subgradient evaluations (i.e. the number of sampling data) for three different types of problems. In all problem instances, we use $\|\cdot\|_2$ norm in both the primal and dual spaces, and hence in the parameter settings of DCS/SDCS $\|\mathbf{L}\|$ refers to the maximum eigenvalue of the Laplacian matrix L.

- **Deterministic convex problems.** The decentralized linear SVM problem under the aforementioned network can be written as

$$\begin{aligned} \min_{\mathbf{x}} \ & \sum_{i=1}^{m} \left[f_i(x_i) := \sum_{(v_j,u_j)\in\mathcal{S}_i} \max\{0, 1 - v_j\langle x_i, u_j\rangle\} + \tfrac{1}{\|\mathcal{S}_i\|}\|x_i\|_1 \right] \\ \text{s.t.} \ & \mathbf{L}\mathbf{x} = \mathbf{0}, \end{aligned} \tag{4.6.2}$$

where $\mathcal{S}_i$ denotes the dataset belonging to node i. Since the problem is deterministic and convex, we choose the parameters of the DCS method as suggested in Theorem 4.3.3, and set the inner iteration limit as $\min(10k, T_k)$ to illustrate the possibility of choosing inner iteration limit dynamically in practice. It needs to be pointed out that if we use a constant inner iteration limit as stated in Theorem 4.3.3, we can obtain similar results as shown in Figure 4.2, but with a slightly slower convergence speed at the very beginning for the DCS method. For distributed dual averaging method, we choose the stepsize in the order of $\mathcal{O}(1/\sqrt{t})$ as suggested in [51].

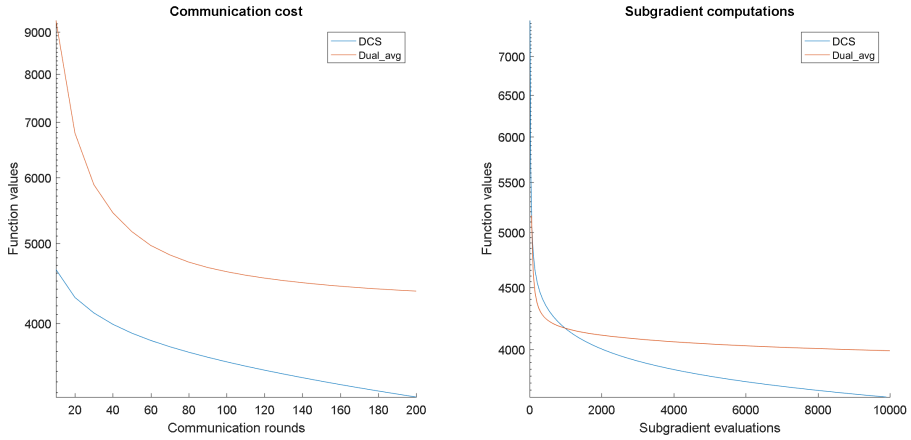

Figure 4.2: The comparison of the DCS method with distributed dual averaging method for solving (4.6.2)

In Figure 4.2, the vertical-axis represents the objective values, the horizontal-axis of the left subgraph is the number of inter-node communication rounds, and the horizontal-axis of the right subgraph is the number of intra-node subgradient evaluations. These numerical results are consistent with our theoretic results in that DCS significantly reduces the total number of inter-node communication rounds while still maintaining comparable bounds on the intra-node subgradient evaluations for solving (4.6.2).

- **Stochastic convex problems.** We now consider a stochastic decentralized linear

SVM problem under the aforementioned network as

$$\min_{\mathbf{x}} \sum_{i=1}^{m} \left[f_i(x_i) := \mathbb{E}_{(v_i,u_i)}[\max\{0, 1 - v_i\langle x_i, u_i\rangle\}] + \frac{1}{\|\mathcal{S}_i\|}\|x_i\|_1 \right] \qquad (4.6.3)$$
$$\text{s.t. } \mathbf{Lx} = \mathbf{0},$$

where (v_i, u_i) represents a uniform random variable with support $\mathcal{S}_i$. For stochastic decentralized communication sliding (SDCS) method, we choose parameters according to Theorem 4.4.8, and also set inner iteration limit as in the deterministic convex case. For distributed dual averaging method, we choose the same stepsize as suggested in [51].

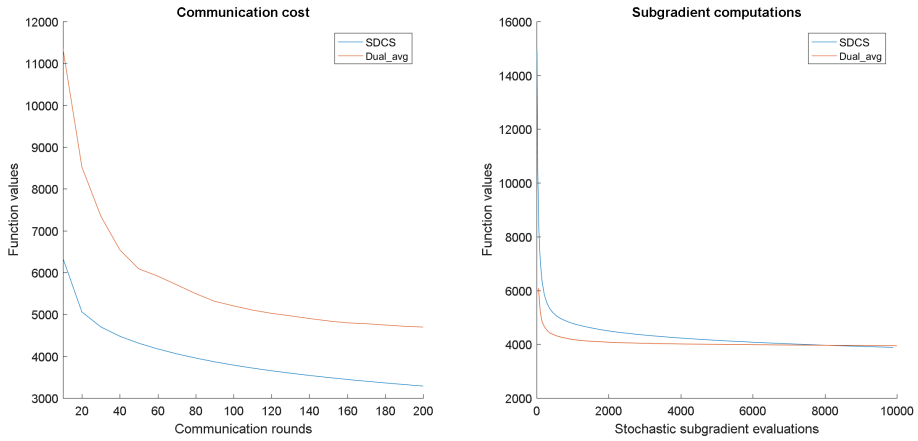

Figure 4.3: The comparison of the SDCS method with distributed dual averaging method for solving (4.6.3)

The above figure clearly shows that SDCS also saves inter-node communication rounds comparing to distributed dual averaging method while preserving the same order of sampling complexity for solving (4.6.3).

- **Stochastic strongly convex problems.** Consider a decentralized linear SVM prob-

lem with l_2 regularizer under the aforementioned network as the following

$$\begin{aligned} \min_{\mathbf{x}} \ & \sum_{i=1}^{m} \left[f_i(x_i) := \mathbb{E}_{(v_i,u_i)}[\max\{0, 1 - v_i\langle x_i, u_i\rangle\}] + \tfrac{1}{2|\mathcal{S}_i|}\|x_i\|_2^2 \right] \\ \text{s.t.} \ & \mathbf{Lx} = \mathbf{0}. \end{aligned} \tag{4.6.4}$$

Since f_i's in (4.6.4) are strongly convex, we choose the parameters of the SDCS method as suggested in Theorem 4.4.10 and fix the inner iteration limit $T_k = 10^4$, which is a rough estimate of the suggested T_k in Theorem 4.4.10. For distributed dual averaging method, we choose stepsize in the order of $\mathcal{O}(1/t)$ as suggested in [115] instead of [51]. This is because [51] did not cover strongly convex problems, while [115] extended the dual averaging method to solve strongly convex problems.

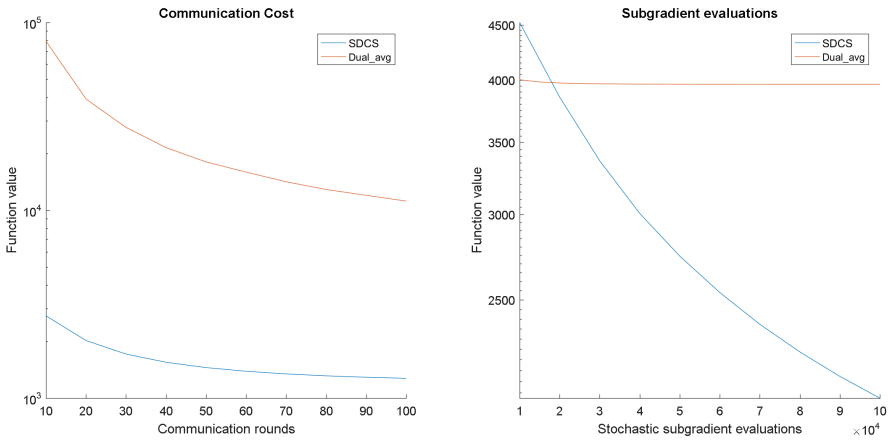

Figure 4.4: The comparison of the SDCS method with distributed dual averaging method for solving (4.6.3)

Figure 4.4 shows that for stochastic strongly convex problems defined in (4.6.4), the SDCS method can achieve better performance than distributed dual averaging method in terms of both the number of communication rounds and subgradient computations. It should be pointed out that SDCS appears to be worse than distributed dual averaging method at the very beginning of the right subgraph because too few

communication rounds are performed by SDCS at that time point, which provides little information about the loss function $F(\mathbf{x})$. However, as the number of communication rounds increases, SDCS gradually outperforms distributed dual averaging method in terms of the objective values. We can also observe similar phenomena in the first two experiments.

In addition to the objective value, we also report the progress of feasibility residual, $\|\mathbf{Lx}\|$, versus communication rounds in Figure 4.5. It needs to be mentioned that the distributed dual averaging method [51] does not measure feasibility residual since it sets the final output to be the average of iterates obtained by one of the network agents[7], and hence requires more rounds of communication to broadcast its final output to all agents, which we do not include in all comparisons.

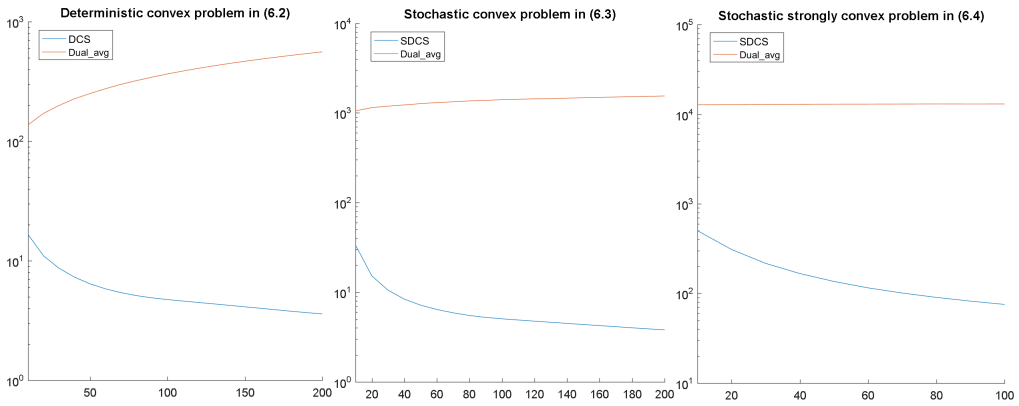

Figure 4.5: The progress of feasibility residuals $\|\mathbf{Lx}\|$ versus communication rounds

4.7 Concluding Remarks of This Chapter

In this chapter, we present a new class of decentralized primal-dual methods which can significantly reduce the number of inter-node communications required to solve the distributed optimization problem in (1.2.13). More specifically, we show that by using these algo-

[7]We choose the average of iterates obtained by the first agent as the output solution for distributed dual averaging method in all three problems.

rithms, the total number of communication rounds can be significantly reduced to $\mathcal{O}(1/\epsilon)$ when the objective functions f_i's are convex and not necessarily smooth. By properly designing the communication sliding algorithms, we demonstrate that the $\mathcal{O}(1/\epsilon)$ number of communications can still be maintained for general convex objective functions (and it can be further reduced to $\mathcal{O}(1/\sqrt{\epsilon})$ for strongly convex objective functions) even if the local subproblems are solved inexactly through iterative procedure (cf. CS procedure) by the network agents. In this case, the number of intra-node subgradient computations that we need will be bounded by $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) when the objective functions f_i's are convex (resp., strongly convex), which is comparable to that required in centralized nonsmooth optimization and not improvable in general. We also establish similar complexity bounds for solving stochastic decentralized optimization counterpart by developing the stochastic communication sliding methods, which can provide communication-efficient ways to deal with streaming data and decentralized statistical inference. As illustrated in our preliminary numerical experiments, all these decentralized communication sliding algorithms have the potential to significantly increase the performance of multiagent systems, where the bottleneck exists in the communication.

CHAPTER 5
ASYNCHRONOUS DECENTRALIZED ACCELERATED STOCHASTIC GRADIENT DESCENT

5.1 Overview

In this chapter, we introduce an asynchronous decentralized accelerated stochastic gradient descent type of method for decentralized stochastic optimization, considering communication and synchronization are the major bottlenecks in decentralized optimization. We establish $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\sqrt{\epsilon})$) communication complexity and $\mathcal{O}(1/\epsilon^2)$ (resp., $\mathcal{O}(1/\epsilon)$) sampling complexity for solving general convex (resp., strongly convex) problems possibly with a composite structure. In particular, we consider decentralized optimization - m agents distributed over the network work collaboratively to solve (1.2.13). Now we assume that $f_i : X_i \to \mathbb{R}$ is a general convex objective function only known to agent i and satisfying

$$\tfrac{\mu}{2}\|x-y\|^2 \le f_i(x) - f_i(y) - \langle f_i'(y), x-y\rangle \le \tfrac{L}{2}\|x-y\|^2 + M\|x-y\|, \ \ \forall x, y \in X_i, \tag{5.1.1}$$

for some $L, M, \mu \ge 0$ and $f_i'(y) \in \partial f_i(y)$, where $\partial f_i(y)$ denotes the subdifferential of f_i at y, and $X_i \subseteq \mathbb{R}^d$ is a closed convex constraint set of agent i. (5.1.1) is a unified way of describing a wide range of problems. In particular, if f_i is a general Lipschitz continuous function with constant M_f, then (5.1.1) holds with $L = 0, \mu = 0$ and $M = 2M_f$. If f_i is a smooth and strongly convex function in $\mathcal{C}^{1,1}_{L/\mu}$ (see [6, Section 1.2.2] for definition), (5.1.1) is satisfied with $M = 0$. Clearly, relation (5.1.1) also holds if f_i is given as the summation of smooth and nonsmooth convex functions. Throughout the chapter, we assume the feasible set X is nonempty.

This chapter is organized as follows. We present our main asynchronous decentralized

primal-dual framework and establish their convergence properties in Section 5.2. Section 5.3 is devoted to providing some preliminary numerical results to demonstrate the advantages of our proposed algorithms. Some technical proofs that support the main theorems in Section 5.2 are provided in Appendix A.

5.1.1 Notation and Terminologies.

We denote by $\mathbf{0}$ and $\mathbf{1}$ the vector of all zeros and ones whose dimensions vary from the context. The cardinality of a set S is denoted by $|S|$. We use I_d to denote the identity matrix in $\mathbb{R}^{d\times d}$. We use $A \otimes B$ for matrices $A \in \mathbb{R}^{n_1\times n_2}$ and $B \in \mathbb{R}^{m_1\times m_2}$ to denote their Kronecker product of size $\mathbb{R}^{n_1m_1\times n_2m_2}$. For a matrix $A \in \mathbb{R}^{n\times m}$, we use $A_{i,j}$ to denote the entry of i-th row and j-th column. For any $m \geq 1$, the set of integers $\{1,\ldots,m\}$ is denoted by $[m]$. We adopt most of the problem setup and reformulation from Section 4.2 of Chapter 4, however, we slightly modify the definition of a stochastic ϵ-solution.

Definition 2 *A point $\hat{\mathbf{x}} \in X^m$ is called a stochastic ϵ-solution of* (1.2.19) *if*

$$\mathbb{E}[F(\hat{\mathbf{x}}) - F(\mathbf{x}^*)] \leq \epsilon \text{ and } \mathbb{E}[\|\mathbf{L}\hat{\mathbf{x}}\|] \leq \epsilon. \tag{5.1.2}$$

We say that $\hat{\mathbf{x}}$ has primal residual ϵ and feasibility residual ϵ.

Note that for problem (1.2.19), the feasibility residual $\|\mathbf{L}\hat{\mathbf{x}}\|$ measures the disagreement among the local copies $\hat{x}_i$, for $i \in \mathcal{N}$. We will use these two criteria to evaluate the output solutions of the algorithms proposed in this chapter. Moreover, the prox-function we refer to in this chapter is defined as in Section 4.2.2 of Chapter 4.

5.2 The Algorithms

In this section, we introduce an asynchronous decentralized primal-dual framework for solving (1.2.13) in the decentralized setting. Specifically, two asynchronous methods are presented, namely asynchronous decentralized primal-dual method in Subsection 5.2.1

and asynchronous accelerated stochastic decentralized communication sliding in Subsection 5.2.2, respectively. Moreover, we establish complexity bounds (number of inter-node communication rounds and/or intra-node stochastic (sub)gradient evaluations) separately in terms of primal functional optimality gap and constraint (or consistency) violation for solving (1.2.13).

5.2.1 Asynchronous Decentralized Primal-dual Method

Our main goals in this subsection are to introduce the basic scheme of asynchronous decentralized primal-dual (ADPD) method, as well as establishing its complexity results. Throughout this subsection, we assume that f_i is a simple function such that we can solve the primal subproblem (5.2.8) explicitly. We will later relax this assumption in Subsection 5.2.2.

We formally present the ADPD method in Algorithm 10. Each agent i maintains two local sequences, namely, the primal estimates $\{x_i^k\}$ and the dual variables $\{y_i^k\}$. All primal estimates x_i^{-1} and x_i^0 are locally initialized from some arbitrary point in X_i, and each dual variable $y_i^0 = \mathbf{0}$. At each iteration $k \geq 1$, only one randomly selected agent (cf. activated agent) $i_k \in [m]$ updates its dual variable $y_{i_k}^k$, and then one randomly selected agent $j_k \in [m]$ updates its primal variable $x_{j_k}^k$. In particular, each agent in the activated agent's neighborhood, i.e., agents $i \in N_{i_k}$, computes a local prediction $\tilde{x}_i^k$ using the two previous primal estimates (ref. (5.2.3)), and send it to agent i_k. In (5.2.4)-(5.2.5), the activated agent i_k calculates its neighborhood disagreement $v_{i_k}^k$ using the receiving messages, and updates the dual variable $y_{i_k}^k$. Other agents' dual variables remain unchanged. Then, another round of communication (5.2.7) between the activated agent j_k and its neighboring agents occurs after the dual prediction step (5.2.6). Lastly, the activated agent j_k solves the proximal projection subproblem (5.2.8) to update $x_{j_k}^k$, and other agents' primal estimates remain the same as the last iteration.

It should be emphasized that each iteration k only involves two communication rounds

(cf. (5.2.4) and (5.2.7)) between the activated agent and its neighboring agents, which significantly reduces synchronous delays appearing in many decentralized methods (e.g., [51, 62, 56] and the proposed methods in Chapter 4), since these methods require at least one communication round between all agents and their neighboring agents iteratively. Also note that similar to the asynchronous ADMM proposed in [64], ADPD employs node-based activation. However, while [64] requires all agents to update dual variables iteratively based on the information obtaining from communication, only the activated agent i_k needs to collect neighboring information and update its dual variable in ADPD (see (5.2.4) and (5.2.5)), and hence ADPD further reduces communication costs and synchronous delays comparing to [64]. Moreover, ADPD can achieve the same rate of convergence $\mathcal{O}(1/\epsilon)$ as [64] under the assumption that (5.2.8) can be solved explicitly. We will demonstrate later that by exploiting the strong convexity, an improved $\mathcal{O}(1/\sqrt{\epsilon})$ rate of convergence can be obtained.

Algorithm 10 Asynchronous decentralized primal-dual (ADPD) update for each agent i

Let $x_i^0 = x_i^{-1} \in X_i$ and $y_i^0 = \mathbf{0}$ for $i \in [m]$, the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$ be given.

for $k = 1, \ldots, N$ **do**

Uniformly choose $i_k, j_k \in [m]$, and update (x_i^k, y_i^k) according to

$$\tilde{x}_i^k = \alpha_k(x_i^{k-1} - x_i^{k-2}) + x_i^{k-1}. \tag{5.2.3}$$

$$v_{i_k}^k = \textstyle\sum_{j \in N_{i_k}} \mathcal{L}_{i_k,j} \tilde{x}_j^k. \tag{5.2.4}$$

$$y_i^k = \begin{cases} \operatorname{argmin}_{y_i \in \mathbb{R}^d} \langle -v_i^k, y_i \rangle + \frac{\tau_k}{2}\|y_i - y_i^{k-1}\|^2 = y_i^{k-1} + \frac{1}{\tau_k} v_i^k, & i = i_k, \\ y_i^{k-1}, & i \neq i_k. \end{cases} \tag{5.2.5}$$

$$\tilde{y}_i^k = m(y_i^k - y_i^{k-1}) + y_i^{k-1}. \tag{5.2.6}$$

$$w_{j_k}^k = \textstyle\sum_{j \in N_{j_k}} \mathcal{L}_{j_k,j} \tilde{y}_j^k. \tag{5.2.7}$$

$$x_i^k = \begin{cases} \operatorname{argmin}_{x_i \in X_i} \langle w_i^k, x_i \rangle + f_i(x_i) + \eta_k V_i(x_i^{k-1}, x_i), & i = j_k, \\ x_j^{k-1}, & i \neq j_k. \end{cases} \tag{5.2.8}$$

end for

For any given weight sequence $\{\hat{\theta}_k\}$ such that $\hat{\theta}_k \geq 0, \quad \sum_{k=0}^N \hat{\theta}_k = 1$, let $\{\theta_k\}$ be

defined as

$$\theta_k = \begin{cases} \hat{\theta}_0 - (m-1)\hat{\theta}_1, & k = 0, \\ m\hat{\theta}_k - (m-1)\hat{\theta}_{k+1}, & k = 1, \ldots, N-1, \\ m\hat{\theta}_N & k = N. \end{cases} \tag{5.2.9}$$

Therefore, $\sum_{k=0}^{N} \theta_k = \sum_{k=0}^{N} \hat{\theta}_k = 1$. In the following theorem, we provide a specific selection of $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$, which leads to $\mathcal{O}(1/\epsilon)$ complexity bounds for the functional optimality gap and also the feasibility residual to obtain a stochastic ϵ-solution of (1.2.19).

Theorem 5.2.1 *Let $\mathbf{x}^*$ be an optimal solution of* (1.2.19)*, and d_{max} be the maximum degree of graph $\mathcal{G}$, and suppose that $\{\alpha_k\}$, $\{\tau_k\}$ and $\{\eta_k\}$ are set to*

$$\alpha_k = m, \ \eta_k = 2md_{max}, \ \textit{and} \ \tau_k = 2md_{max}, \forall k = 1, \ldots, N. \tag{5.2.10}$$

Then, for any $N \geq 1$, we have

$$\mathbb{E}_{[i_k, j_k]}\{F(\bar{\mathbf{x}}^N) - F(\mathbf{x}^*)\} \leq \mathcal{O}\left\{\tfrac{m\Delta_{\mathbf{x}^0}}{N+m}\right\}, \ \ \mathbb{E}_{[i_k, j_k]}\{\|\mathbf{L}\bar{\mathbf{x}}^N\|\} \leq \mathcal{O}\left\{\tfrac{m\Delta_{\mathbf{x}^0}}{N+m}\right\}, \tag{5.2.11}$$

where $\bar{\mathbf{x}}^N = \frac{1}{N+m}(\sum_{k=0}^{N-1}\mathbf{x}^k + m\mathbf{x}^N)$, $\{\mathbf{x}^k\}$ is generated by Algorithm 10, and $\Delta_{\mathbf{x}^0} := \max\left\{C_{\mathbf{x}^0}, \|\mathbf{L}\mathbf{x}^0\| + md_{max}\left(\|\mathbf{y}^*\| + \sqrt{\frac{C_{\mathbf{x}^0} + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{md_{max}}}\right)\right\}$ with $C_{\mathbf{x}^0} = F(\mathbf{x}^0) - F(\mathbf{x}^*) + md_{max}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$.

Proof. Let us set $\{\hat{\theta}_k\}$ as follow

$$\hat{\theta}_k = \begin{cases} \frac{m}{N+m}, & k = 0, \\ \frac{1}{N+m}, & k = 1, \ldots, N. \end{cases} \tag{5.2.12}$$

Therefore, it is easy to check that (5.2.10) satisfies conditions (A.0.10)-(A.0.13) in Propo-

sition A.0.4. Also note that by (5.2.9), we have

$$\theta_k = \begin{cases} \frac{1}{N+m}, & k = 1, \ldots, N-1, \\ \frac{m}{N+m}, & k = N, \end{cases} \tag{5.2.13}$$

which implies that $\bar{\mathbf{x}}^N = \frac{1}{N+m}(\sum_{k=0}^{N-1} \underline{\mathbf{x}}^k + m\underline{\mathbf{x}}^N)$. By plugging the parameter setting in (A.0.14), we have

$$\mathbb{E}_{[i_k,j_k]}\{Q(\bar{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y})\} \leq \frac{m}{N+m}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + 2md_{max}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)\right] + \mathbb{E}_{[i_k,j_k]}\{\langle \mathbf{s}, \mathbf{y}\rangle\}. \tag{5.2.14}$$

Observe that from (A.0.15) and (5.2.10),

$$\mathbb{E}_{[i_k,j_k]}\{\|\mathbf{s}\|\} \leq \frac{m}{N+m}\mathbb{E}_{[i_k,j_k]}\left[\|\mathbf{L}\mathbf{x}^0\| + 2d_{max}\|x_{j_N}^N - x_{j_N}^{N-1}\| + 2md_{max}(\|\mathbf{y}^* - \mathbf{y}^N\| + \|\mathbf{y}^*\|)\right].$$

By (A.0.16), (5.2.10) and Jensen's inequality, we have

$$(\mathbb{E}\{\|x_{j_N}^N - x_{j_N}^{N-1}\|\})^2 \leq \mathbb{E}\{\|x_{j_N}^N - x_{j_N}^{N-1}\|^2\} \leq 4\left[\frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{md_{max}} + 2\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^*\|^2\right],$$
$$(\mathbb{E}\{\|\mathbf{y}^* - \mathbf{y}^N\|\})^2 \leq \mathbb{E}\{\|\mathbf{y}^* - \mathbf{y}^N\|^2\} \leq 2\left[\frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{md_{max}} + 2\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^*\|^2\right].$$

Hence, in view of the above three inequalities, we conclude that

$$\begin{aligned}\mathbb{E}_{[i_k,j_k]}\{\|\mathbf{s}\|\} &\leq \frac{m}{N+m}\Big\{\|\mathbf{L}\mathbf{x}^0\| + 2md_{max}\|\mathbf{y}^*\| \\ &\quad + 7md_{max}\sqrt{\tfrac{F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{md_{max}} + 2\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \|\mathbf{y}^*\|^2}\Big\} \\ &= \mathcal{O}\left\{\frac{m}{N+m}\left[\|\mathbf{L}\mathbf{x}^0\| + md_{max}\|\mathbf{y}^*\| + md_{max}\sqrt{\tfrac{F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{md_{max}} + \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)}\right]\right\}.\end{aligned}$$

Furthermore, by (5.2.14) we have

$$\mathbb{E}_{[i_k,j_k]}\{g(\mathbf{s}, \bar{\mathbf{z}}^N)\} \leq \frac{m}{N+m}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + 2md_{max}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)\right].$$

The results in (5.2.11) immediately follow from Proposition 4.2.1 and the above two inequalities. ∎

Theorem 5.2.1 implies the total number of inter-node communication rounds performed by ADPD to find a stochastic ϵ-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\frac{md_{max}\Delta_{\mathbf{x}^0}}{\epsilon}\right\}. \tag{5.2.15}$$

Observed that in Algorithm 10, we assume that f_i's are simple functions such that (5.2.8) can be solved explicitly. However, since f_i's are possibly nonsmooth functions and/or possess composite structures, it is often difficult to solve (5.2.8) especially when f_i is provided in the form of expectation. In the next subsection, we present a new asynchronous stochastic decentralized primal-dual type method, called the asynchronous accelerated stochastic decentralized communication sliding (AA-SDCS) method, for the case when (5.2.8) is not easy to solve.

5.2.2 Asynchronous Accelerated Stochastic Decentralized Communication Sliding

In the subsection, we show that one can still maintain the same number of inter-node communications even when the subproblem (5.2.8) is approximately solved through an optimal stochastic approximation method, namely AC-SA proposed in [116, 112, 10], and that the total number of required stochastic (sub)gradient evaluations (or sampling complexity) is comparable to centralized mirror descent methods. Throughout this subsection, we assume that only noisy (sub)gradient information of f_i, $i = 1, \ldots, m$, is available or easier to compute. This situation happens when the function f_i's are given either in the form of expectation or as the summation of lots of components. Moreover, we assume that the first-order information of the function f_i, $i = 1, \ldots, m$, can be accessed by a stochastic oracle (SO), which, given a point $u^t \in X$, outputs a vector $G_i(u^t, \xi_i^t)$ such that (4.4.1) and (4.4.2) hold. We call $G_i(u^t, \xi_i^t)$ a *stochastic (sub)gradient* of f_i at u^t. Observe that this assumption

covers the case that one can access the exact (sub)gradients of f_i whenever $\sigma = 0$.

We now add a few comments about Algorithm 11. Firstly, similar to SDCS proposed in Chapter 4, AA-SDCS exploits two loops: the doubly randomized primal-dual scheme as outer loop and ACS as inner loop. More specifically, AA-SDCS utilizes the AC-SA method proposed in [116, 112, 10] to approximately solve the primal subproblem in (5.2.8), which provides a unified scheme for solving a general class of problems defined in (5.1.1) and leads to accelerated rates of convergence when f_i possesses smooth structure. Secondly, the same dual information $w = w_{j_k}^k$ (see (5.2.20)) has been used throughout the $T = T_k$ iterations of the ACS procedure, and hence no additional communication is required within the procedure. Finally, since AA-SDCS randomly selects one subproblem (5.2.8) and solved it inexactly, the outer loop also needs to be modified to attain the best possible rate of convergence. In fact, the ACS procedure provides two approximate solutions of (5.2.8): one is the primal estimate $\{x_i^k\}$ and the other is $\{\underline{x}_i^k\}$, which will be maintained by each agent and later plays a crucial role in the development and convergence analysis of AA-SDCS. We also accordingly modify the primal extrapolation step (5.2.16) of the outer loop. For later convenience, we refer to the subproblem ACS solved at iteration k as $\Phi^k(x_i)$, i.e.,

$$\operatorname{argmin}_{x_i \in X_i} \left\{ \Phi^k(x_i) := \langle w_i^k, x_i \rangle + f_i(x_i) + \eta_k V_i(x_i^{k-1}, x_i) \right\}. \tag{5.2.26}$$

The following theorem provides a specific selection of $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{T_k\}$ for Algorithm 11, which leads to $\mathcal{O}(1/\epsilon)$ complexity bounds for the functional optimality gap and also the feasibility residual to obtain a stochastic ϵ-solution of (1.2.19).

Theorem 5.2.2 *Let* $\mathbf{x}^*$ *be an optimal solution of* (1.2.19)*, and* d_{max} *be the maximum degree of graph* $\mathcal{G}$*, and suppose that the parameters* $\{\lambda_t\}$ *and* $\{\beta_t\}$ *in the* ACS *procedure of Algorithm 11 be set to*

$$\lambda_t = \tfrac{2}{t+1},\ \beta_t = \tfrac{4(\mathcal{C}+L)}{t(t+1)},\ \forall t \geq 1, \tag{5.2.27}$$

Algorithm 11 Asynchronous Accelerated Stochastic Decentralized Communication Sliding (AA-SDCS)

Let $x_i^0 = x_i^{-1} = \underline{x}_i^0 \in X_i$, $y_i^0 = \mathbf{0}$ for $i \in [m]$ and the nonnegative parameters $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{T_k\}$ be given.

for $k = 1, \ldots, N$ **do**

Uniformly choose $i_k, j_k \in [m]$, and update $(\underline{x}_i^k, y_i^k)$ according to

$$\tilde{x}_i^k = \alpha_k[m\underline{x}_i^{k-1} - (m-1)\underline{x}_i^{k-2} - x_i^{k-2}] + x_i^{k-1}. \tag{5.2.16}$$

$$v_{i_k}^k = \textstyle\sum_{j \in N_{i_k}} \mathcal{L}_{i_k,j}\tilde{x}_j^k. \tag{5.2.17}$$

$$y_i^k = \begin{cases} y_i^{k-1} + \frac{1}{\tau_k} v_i^k, & i = i_k, \\ y_i^{k-1}, & i \neq i_k. \end{cases} \tag{5.2.18}$$

$$\tilde{y}_i^k = m(y_i^k - y_i^{k-1}) + y_i^{k-1}. \tag{5.2.19}$$

$$w_{j_k}^k = \textstyle\sum_{j \in N_{j_k}} \mathcal{L}_{j_k,j}\tilde{y}_j^k. \tag{5.2.20}$$

$$(x_i^k, \underline{x}_i^k) = \begin{cases} \mathrm{ACS}(f_i, X_i, V_i, T_k, \eta_k, w_i^k, x_i^{k-1}), & i = j_k, \\ (x_i^{k-1}, \underline{x}_i^{k-1}), & i \neq j_k. \end{cases} \tag{5.2.21}$$

end for

The ACS (Accelerated Communication-Sliding) procedure called at (5.2.21) is stated as follows.

procedure: $(x, \underline{x}) = \mathrm{ACS}(\phi, U, V, T, \eta, w, x)$

Let $u^0 = \underline{u}^0 = x$ and the parameters $\{\beta_t\}$ and $\{\lambda_t\}$ be given.

for $t = 1, \ldots, T$ **do**

$$\hat{u}^t = \tfrac{(1-\lambda_t)(\mu+\eta+\beta_t)}{\beta_t+(1-\lambda_t^2)(\mu+\eta)}\underline{u}^{t-1} + \tfrac{\lambda_t[(1-\lambda_t)(\mu+\eta)+\beta_t]}{\beta_t+(1-\lambda_t^2)(\mu+\eta)}u^{t-1}. \tag{5.2.22}$$

$$G^t = G(\hat{u}^t, \xi^t). \tag{5.2.23}$$

$$\begin{aligned} u^t = \mathrm{argmin}_{u \in U} \big\{ &\lambda_t[\langle w + G^t + \eta(\nabla w(\hat{u}^t) - \nabla w(x)), u\rangle + (\mu+\eta)V(\hat{u}^t, u)] \\ &+[(1-\lambda_t)(\mu+\eta) + \beta_t]V(u^{t-1}, u)\big\}. \end{aligned} \tag{5.2.24}$$

$$\underline{u}^t = (1-\lambda_t)\underline{u}^{t-1} + \lambda_t u^t. \tag{5.2.25}$$

end for

Set $x = u^T$ and $\underline{x} = \underline{u}^T$.

end procedure

and $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ *and* $\{T_k\}$ *are set to*

$$\begin{aligned} &\alpha_k = 1,\ \eta_k = 4md_{max},\ \tau_k = 2d_{max}, \\ &\text{and } T_k = \max\left\{\left\lceil \tfrac{(M^2+\sigma^2)N}{d_{max}\mathcal{D}} \right\rceil, \left\lceil \sqrt{\tfrac{\mathcal{C}+L}{md_{max}}} \right\rceil\right\},\ \forall k = 1, \ldots, N, \end{aligned} \tag{5.2.28}$$

for some $\mathcal{D} > 0$. *Then, for any* $N \geq 1$, *we have*

$$\mathbb{E}\{F(\bar{\mathbf{x}}^N) - F(\mathbf{x}^*)\} \leq \mathcal{O}\left\{\tfrac{m\Delta_{\mathbf{x}^0,\mathcal{D}}}{N+m}\right\},\ \ \mathbb{E}\{\|\mathbf{L}\bar{\mathbf{x}}^N\|\} \leq \mathcal{O}\left\{\tfrac{m\Delta_{\mathbf{x}^0,\mathcal{D}}}{N+m}\right\}, \tag{5.2.29}$$

where $\bar{\mathbf{x}}^N = \frac{1}{N+m}(\sum_{k=0}^{N-1}\underline{\mathbf{x}}^k + m\underline{\mathbf{x}}^N)$, $\{\underline{\mathbf{x}}^k\}$ *is generated by Algorithm 11, and* $\Delta_{\mathbf{x}^0,\mathcal{D}} := \max\left\{C_{\mathbf{x}^0,\mathcal{D}}, \|\mathbf{L}\mathbf{x}^0\| + d_{max}\left(\|\mathbf{y}^*\| + \sqrt{\frac{C_{\mathbf{x}^0,\mathcal{D}}+\langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{d_{max}}}\right)\right\}$ *with* $C_{\mathbf{x}^0,\mathcal{D}} = F(\mathbf{x}^0) - F(\mathbf{x}^*) + md_{max}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\mathcal{D}}{m}$.

Proof. Let us set $\{\hat{\theta}_k\}$ as (5.2.12). Therefore, it is easy to check that parameter settings (5.2.27) and (5.2.28) satisfies conditions (A.0.22) - (A.0.24), (A.0.9), and (A.0.27) - (A.0.30) (cf. Proposition A.0.5 and A.0.6 Appendix A). Also note that by (5.2.9), $\{\theta_k\}$ is given by (5.2.13), which implies that $\bar{\mathbf{x}}^N = \frac{1}{N+m}(\sum_{k=0}^{N-1}\underline{\mathbf{x}}^k + m\underline{\mathbf{x}}^N)$. By plugging the parameter setting in (A.0.31), we have

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y})\} \leq \tfrac{m}{N+m}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + 8md_{max}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{2\mathcal{D}}{m}\right] + \mathbb{E}\{\langle \mathbf{s}, \mathbf{y}\rangle\}. \tag{5.2.30}$$

Observe that from (A.0.32) and (5.2.28)

$$\mathbb{E}\{\|\mathbf{s}\|\} \leq \tfrac{m}{N+m}\mathbb{E}\left[\|\mathbf{L}\mathbf{x}^0\| + \tfrac{\|\mathbf{L}\|}{m}\|\hat{\underline{\mathbf{x}}}^N - \mathbf{x}^{N-1}\| + 2d_{max}(\|\mathbf{y}^* - \mathbf{y}^N\| + \|\mathbf{y}^*\|)\right].$$

By (A.0.33), (5.2.28), and Jensen's inequality, we have

$$(\mathbb{E}\{\|\hat{\underline{\mathbf{x}}}^N - \mathbf{x}^{N-1}\|\})^2 \leq \mathbb{E}\{\|\hat{\underline{\mathbf{x}}}^N - \mathbf{x}^{N-1}\|^2\}$$

$$\leq \tfrac{2(F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle)}{d_{max}} + 16m\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + 2\|\mathbf{y}^*\|^2 + \tfrac{4\mathcal{D}}{md_{max}},$$

$$(\mathbb{E}\{\|\mathbf{y}^* - \mathbf{y}^N\|\})^2 \leq \mathbb{E}\{\|\mathbf{y}^* - \mathbf{y}^N\|^2\}$$

$$\leq \tfrac{2(F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle)}{d_{max}} + 16m\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + 2\|\mathbf{y}^*\|^2 + \tfrac{4\mathcal{D}}{md_{max}}.$$

Hence, in view of the above three inequalities, we obtain

$$\begin{aligned}\mathbb{E}\{\|\mathbf{s}\|\} &\leq \tfrac{m}{N+m}\Big\{\|\mathbf{L}\mathbf{x}^0\| + 2d_{max}\|\mathbf{y}^*\| \\ &\quad + 3d_{max}\sqrt{\tfrac{2(F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle)}{d_{max}} + 16m\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + 2\|\mathbf{y}^*\|^2 + \tfrac{4\mathcal{D}}{md_{max}}}\Big\} \\ &= \mathcal{O}\left\{\tfrac{m}{N+m}\left[\|\mathbf{L}\mathbf{x}^0\| + d_{max}\|\mathbf{y}^*\| + d_{max}\sqrt{\tfrac{F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle \mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle}{d_{max}} + m\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \tfrac{\mathcal{D}}{md_{max}}}\right]\right\}.\end{aligned}$$

Furthermore, by (5.2.30) we have

$$\mathbb{E}\{g(\mathbf{s},\bar{\mathbf{z}}^N)\} \leq \tfrac{m}{N+m}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + 8md_{max}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \tfrac{2\mathcal{D}}{m}\right].$$

The results in (5.2.29) immediately follow from applying Proposition 4.2.1 to the above two inequalities. ■

In view of Theorem 5.2.2, letting $\mathcal{D} = \mathcal{O}(m^2 d_{max})$, we can see that the total number of inter-node communication rounds and intra-node (sub)gradient evaluations required by AA-SDCS for finding a stochastic ϵ-solution of (1.2.19) can be bounded by

$$\mathcal{O}\left\{\tfrac{md_{max}\Delta_{\mathbf{x}^0,\mathcal{D}}}{\epsilon}\right\} \text{ and } \mathcal{O}\left\{\tfrac{(M^2+\sigma^2)\Delta^2_{\mathbf{x}^0,\mathcal{D}}}{\epsilon^2 d^2_{max}} + \sqrt{\tfrac{m(\mathcal{C}+L)}{d_{max}}}\tfrac{\Delta_{\mathbf{x}^0,\mathcal{D}}}{\epsilon}\right\}, \tag{5.2.31}$$

respectively. It also needs to be emphasized that the sampling complexity (second bound in (5.2.31) only sublinearly depends on the Lipschitz constant L.

Now consider the case when f_i's are strongly convex (i.e., $\mu > 0$ in (5.1.1)). In order to exploit the strong convexity, we assume that the prox-functions $V_i(\cdot,\cdot)$ (cf. (4.2.7)) are growing quadratically with the *quadratic growth constant* $\mathcal{C}$, i.e., there exists a constant

$\mathcal{C} > 0$ such that (4.3.32) holds. By (4.2.8), we must have $\mathcal{C} \geq 1$. The following theorem instantiates Algorithm 11 by providing a selection of $\{\alpha_k\}$, $\{\tau_k\}$, $\{\eta_k\}$ and $\{T_k\}$, which leads to a improved $\mathcal{O}(1/\sqrt{\epsilon})$ complexity bound for the functional optimality gap and also the feasibility residual to obtain a stochastic ϵ-solution of (1.2.19).

Theorem 5.2.3 *Let* $\mathbf{x}^*$ *be an optimal solution of* (1.2.19)*, and* d_{max} *be the maximum degree of graph* $\mathcal{G}$*, and suppose that the parameters* $\{\lambda_t\}$ *and* $\{\beta_t\}$ *in the* ACS *procedure of Algorithm 11 be set to* (5.2.27)*, and* $\{\alpha_k\}$*,* $\{\tau_k\}$*,* $\{\eta_k\}$ *and* $\{T_k\}$ *are set to*

$$\begin{aligned} &\alpha_k = \tfrac{k+3m-1}{k+3m},\ \eta_k = \tfrac{(k+3m-1)\mu}{2} - \tfrac{\mathcal{C}+L}{T_k(T_k+1)},\ \tau_k = \tfrac{32md_{max}^2}{(k+3m)\mu}, \\ &\text{and } T_k = \max\left\{\left\lceil \tfrac{64m(M^2+\sigma^2)N}{\mathcal{D}\mu^2}\right\rceil, \left\lceil\sqrt{\tfrac{4(\mathcal{C}+L)}{(k+3m-3)\mu}}\right\rceil\right\},\ \forall k = 1,\ldots,N. \end{aligned} \tag{5.2.32}$$

Then, for any $N \geq 1$*, we have*

$$\mathbb{E}\{F(\bar{\mathbf{x}}^N) - F(\mathbf{x}^*)\} \leq \mathcal{O}\left\{\tfrac{m^2\Delta_{\mathbf{x}^0,\mathcal{D},\mu}}{m^2+N^2}\right\},\ \ \mathbb{E}\{\|\mathbf{L}\bar{\mathbf{x}}^N\|\} \leq \mathcal{O}\left\{\tfrac{m^2\Delta_{\mathbf{x}^0,\mathcal{D},\mu}}{m^2+N^2}\right\}, \tag{5.2.33}$$

where $\bar{\mathbf{x}}^N = \frac{2}{6m^2+N(N+6m+1)}(\sum_{k=0}^{N-1}(k+2m+1)\underline{\mathbf{x}}^k + m(N+3m)\underline{\mathbf{x}}^N)$*,* $\{\underline{\mathbf{x}}^k\}$ *is generated by Algorithm 11, and* $\Delta_{\mathbf{x}^0,\mathcal{D},\mu} := \max\left\{C_{\mathbf{x}^0,\mathcal{D},\mu}, \|\mathbf{L}\mathbf{x}^0\| + \frac{d_{max}^2\|\mathbf{y}^*\|}{\mu} + d_{max}\sqrt{\frac{C_{\mathbf{x}^0,\mathcal{D},\mu}+\langle\mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle}{\mu}}\right\}$ *with* $C_{\mathbf{x}^0,\mathcal{D},\mu} = F(\mathbf{x}^0) - F(\mathbf{x}^*) + m\mu\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \frac{\mathcal{D}\mu}{m^2}$.

Proof. Let us set

$$\hat{\theta}_k = \begin{cases} \frac{6m^2}{6m^2+N(N+6m+1)}, & k = 0, \\ \frac{2(k+3m)}{6m^2+N(N+6m+1)}, & k = 1,\ldots,N. \end{cases} \tag{5.2.34}$$

Observe from (5.2.32) that

$$\eta_k = \tfrac{(k+3m-1)\mu}{2} - \tfrac{\mathcal{C}+L}{T_k(T_k+1)} \geq \tfrac{(k+3m-1)\mu}{2} - \tfrac{(k+3m-3)\mu}{4} = \tfrac{(k+3m+1)\mu}{4}.$$

Therefore, it is easy to check that parameter settings (5.2.27) and (5.2.32) satisfy conditions

(A.0.22) - (A.0.24), (A.0.9), (A.0.28) - (A.0.30), and (A.0.39). Also by (5.2.9), we have

$$\theta_k = \begin{cases} \frac{2(k+2m+1)}{6m^2+N(N+6m+1)}, & k = 1, \ldots, N-1, \\ \frac{2m(N+3m)}{6m^2+N(N+6m+1)}, & k = N, \end{cases}$$

which implies that $\bar{\mathbf{x}}^N = \frac{2}{6m^2+N(N+6m+1)}(\sum_{k=0}^{N-1}(k+2m+1)\underline{\mathbf{x}}^k + m(N+3m)\underline{\mathbf{x}}^N)$. By plugging the parameter setting in (A.0.40), we have

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N; \mathbf{x}^*, \mathbf{y})\} \leq \frac{6m^2}{6m^2+N(N+6m+1)}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + \frac{(3m+1)\mu}{2}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{\mathcal{D}\mu}{6m^2}\right] + \mathbb{E}\{\langle \mathbf{s}, \mathbf{y}\rangle\}. \tag{5.2.35}$$

Observe that from (A.0.32) and (5.2.32)

$$\mathbb{E}\{\|\mathbf{s}\|\} \leq \frac{2m^2}{6m^2+N(N+6m+1)}\mathbb{E}\left[3\|\mathbf{L}\mathbf{x}^0\| + \frac{(N+3m)\|\mathbf{L}\|}{m^2}\|\hat{\underline{\mathbf{x}}}^N - \mathbf{x}^{N-1}\| + \frac{32d_{max}^2}{\mu}(\|\mathbf{y}^* - \mathbf{y}^N\| + \|\mathbf{y}^*\|)\right].$$

In view of (A.0.41) and (5.2.32), we have

$$\begin{aligned}
\mathbb{E}\{\|\hat{\underline{\mathbf{x}}}^N - \mathbf{x}^{N-1}\|^2\} &\leq \frac{8}{\hat{\theta}_N\eta_N}\frac{6m^2}{6m^2+N(N+6m+1)}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle + \frac{(3m+1)\mu}{2}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)\right. \\
&\quad \left. + \frac{32d_{max}^2}{3\mu}\|\mathbf{y}^*\|^2 + \frac{\mathcal{D}\mu}{6m^2}\right] \\
&\leq \frac{96m^2}{(N+3m)(N+3m+1)}\left[\frac{F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{\mu} + \frac{(3m+1)}{2}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)\right. \\
&\quad \left. + \frac{32d_{max}^2}{3\mu^2}\|\mathbf{y}^*\|^2 + \frac{\mathcal{D}}{6m^2}\right], \\
\mathbb{E}\{\|\mathbf{y}^* - \mathbf{y}^N\|^2\} &\leq \frac{3\mu^2}{4d_{max}^2}\left[\frac{F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle}{\mu} + \frac{(3m+1)}{2}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{32d_{max}^2}{3\mu^2}\|\mathbf{y}^*\|^2 + \frac{\mathcal{D}}{6m^2}\right].
\end{aligned}$$

Hence, in view of the above three inequalities and Jensen's inequality, we obtain

$$\begin{aligned}
\mathbb{E}\{\|\mathbf{s}\|\} \leq \frac{2m^2}{6m^2+N(N+6m+1)}\Bigg\{&3\|\mathbf{L}\mathbf{x}^0\| + \frac{32d_{max}^2}{\mu}\|\mathbf{y}^*\| \\
&+ 24d_{max}\sqrt{\frac{3(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle)}{\mu} + \frac{3(3m+1)}{2}\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \frac{32d_{max}^2}{\mu^2}\|\mathbf{y}^*\|^2 + \frac{\mathcal{D}}{2m^2}}\Bigg\}
\end{aligned}$$

$$= \mathcal{O}\left\{\frac{m^2}{m^2+N^2}\left[\|\mathbf{L}\mathbf{x}^0\| + \frac{d_{max}^2}{\mu}\|\mathbf{y}^*\| + d_{max}\sqrt{\frac{(F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle\mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle)}{\mu} + m\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \frac{\mathcal{D}}{m^2}}\right]\right\}.$$

Furthermore, by (5.2.35) we have

$$\mathbb{E}\{g(\mathbf{s},\bar{\mathbf{z}}^N)\} \leq \frac{6m^2}{6m^2+N(N+6m+1)}\left[F(\mathbf{x}^0) - F(\mathbf{x}^*) + \frac{(3m+1)\mu}{2}\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \frac{\mathcal{D}\mu}{6m^2}\right].$$

The results in (5.2.33) immediately follow from applying Proposition 4.2.1 to the above two inequalities. ■

As a consequence of Theorem 5.2.3, letting $\mathcal{D} = \mathcal{O}(m^3)$, we can see that the total number of inter-node communication rounds and intra-node (sub)gradient evaluations required by AA-SDCS for finding a stochastic ϵ-solution of (1.2.19), respectively, can be bounded by

$$\mathcal{O}\left\{md_{max}\sqrt{\frac{\Delta_{\mathbf{x}^0,\mathcal{D},\mu}}{\epsilon}}\right\}, \text{ and } \mathcal{O}\left\{\frac{(M^2+\sigma^2)\Delta_{\mathbf{x}^0,\mathcal{D},\mu}}{\mu^2\epsilon} + \sqrt{\frac{m(\mathcal{C}+L)}{\mu}}\left(\frac{\Delta_{\mathbf{x}^0,\mathcal{D},\mu}}{\epsilon}\right)^{1/4}\right\}. \quad (5.2.36)$$

5.3 Numerical Experiments

We demonstrate the advantages of our proposed AA-SDCS method over the state-of-the-art synchronous algorithm, stochastic decentralized communication sliding (SDCS) method, proposed in Chapter 4 through some preliminary numerical experiments.

Let us consider the decentralized linear Support Vector Machines (SVM) model with the following hinge loss function

$$\max\{0, 1 - v\langle x, u\rangle\},$$

where $(v, u) \in \mathbb{R} \times \mathbb{R}^d$ is the pair of class label and feature vector, and $x \in \mathbb{R}^d$ denotes the weight vector. We consider two types of stochastic decentralized linear SVM problems in this chapter. For the convex case, we study 1-norm SVM problem [113, 114], i.e., the hinge loss function defined above plus $l1$-norm as the regularizer, formulated as in (4.6.3), while

for the strongly convex case, we study 2-norm SVM model, formulated as in (4.6.4). Moreover, we use the Erhos-Renyi algorithm [1] to generate the underlying decentralized network. Note that nodes with different degrees are drawn in different colors (cf. Figure5.1). We also used the real dataset named "ijcnn1" from LIBSVM[2] and drew $40,000$ samples from this dataset as our problem instance data to train the decentralized linear SVM model. These samples are evenly split over the network agents. For example, if we have $m = 8$ nodes (or agents) in the decentralized network (see Figure 5.1), each network agent has $5,000$ samples.

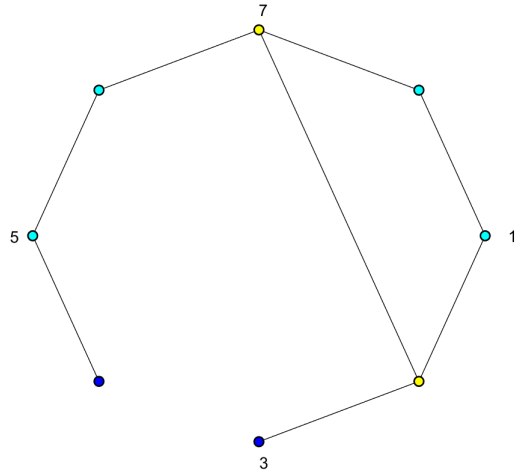

Figure 5.1: The 8-agent decentralized network randomly generated by Erhos-Renyi algorithm.

With the same initial points $\mathbf{x}^0 = \mathbf{1}$ and $\mathbf{y}^0 = \mathbf{0}$, we compare the performances of our algorithms with the SDCS method for solving (1.2.13)-(1.2.19) by showing the progresses of objective function values and feasibility residuals $\|\mathbf{Lx}\|$ versus the elapsed CPU running time (in seconds) for the aforementioned two different types of problems. In all problem instances, we use $\|\cdot\|_2$ norm in both the primal and dual spaces, and hence in the parameter settings of SDCS $\|\mathbf{L}\|$ refers to the maximum eigenvalue of the Laplacian matrix $\mathcal{L}$.

[1] We implemented the Erhos-Renyi algorithm based on a MATLAB function written by Pablo Blider, which can be found in `https://www.mathworks.com/matlabcentral/fileexchange/4206`.

[2] This real dataset can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Moreover, all algorithms are implemented in MATLAB R2016a and run in the computer environment of with 32-core (Intel(R) Xeon(R) CPU E5-2673 v3 2.40GHz) virtual machine on Microsoft Azure. We then utilize the parallel toolbox in MATLAB to simulate the synchronous setting for the SDCS method. However, inter-node communication is instant and no delay is simulated in all experiments. In fact, such simulation setup is in favor of the synchronous methods, since these methods can be heavily slowed down by different processing speeds of the agents (cores) and inter-node communication speeds.

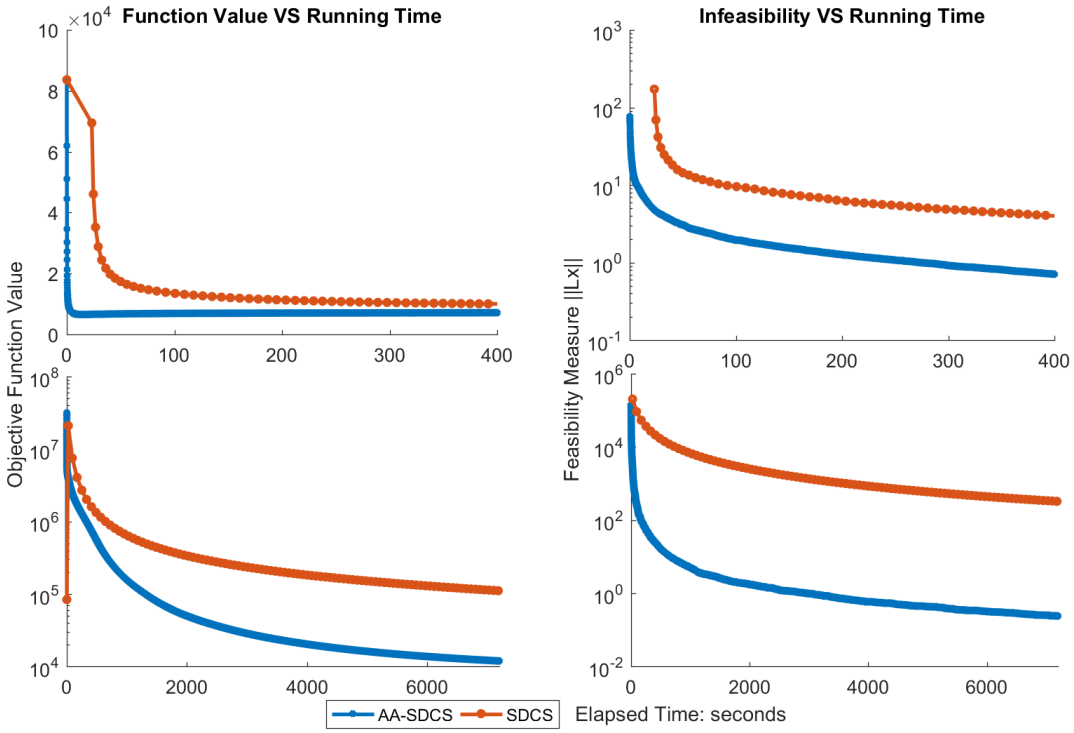

Figure 5.2: Convergence comparison for solving decentralized 1-norm SVM (cf. first row) and decentralized 2-norm SVM (cf. second row) defined over the connected decentralized network with 8 agents (cf. Figure 5.1).

The above figure clearly shows that for all testing instances, AA-SDCS can significantly save CPU running time over SDCS in terms of both objective function values and feasibility residuals. Moreover, AA-SDCS has greater improvements over SDCS for solving decentralized 2-norm SVM problems, which is consistent with our theoretic results that AA-SDCS achieves acceleration when the objective function contains a smooth component.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In this thesis we study first-order methods for distributed optimization and machine learning problems. Our main goal is to design and analyze efficient stochastic first-order algorithms for solving distributed convex optimization problems. In particular, inspired by interpreting an optimal deterministic first-order method, the Nesterov's accelerated gradient (NAG) method, we designed optimal randomized methods for solving the federated learning problems, a.k.a. server-worker distributed optimization problems; We can summarize the main contributions as follows:

- provided a natural iterative buyer-supplier game interpretation for the well-known Nesterov's accelerated gradient method, which is often difficult to interpret, and developed a new optimal deterministic first-order method, the primal-dual gradient (PDG) method. PDG covers a variant of the NAG method as a special case;

- developed an optimal randomized incremental gradient (RIG) method, namely randomized primal-dual gradient (RPDG) method, for solving the finite-sum optimization problems, which is the core problem class in distributed optimization. RPDG only requires an one-time exact gradient computation at the initial point and one gradient evaluation of the randomly selected component function (cf. local loss function of one network agent) iteratively, which leads to significant savings of gradient computations and communication costs in the distributed setting;

- established a lower complexity bound for the class of RIG method for solving the finite-sum optimization problems, which matched the upper complexity bound given by RPDG and hence proved the optimality of RPDG. As a byproduct, we also derived a lower complexity bound for randomized block coordinate descent methods

by utilizing the separable structure of the aforementioned worst-case instances.

- further improved RPDG by introducing a new optimal RIG method, namely the randomized gradient extrapolation method (RGEM), to solve a much broader class of finite-sum optimization problems. Without any full gradient computation, RGEM possesses iteration costs as low as pure stochastic gradient descent (SGD) methods, but achieves a much faster and optimal linear rate of convergence for solving deterministic finite-sum problems. Moreover, RGEM can be interpreted as the randomized version of the dual of the NAG method, which utilizes a gradient extrapolation step for estimating exact gradients in addition to predicting dual information;

- extended RGEM to stochastic finite-sum optimization, and established the sampling complexity bound that is independent of m, the number of agents, and the communication complexity that is only linearly depends on m or even $\sqrt{m}$ for ill-conditioned problems.

Moreover, we investigated another type of distributed optimization problems, the decentralized optimization, where the underlying distributed network does not have a central authority. Consider communication is the major bottleneck, we provided a theoretical understanding on how many rounds of inter-node communications and intra-node (stochastic) (sub)gradient computations are required in order to solve the decentralized problems to certain accuracy in which the objective functions are convex or strongly convex, but not necessarily smooth, and their exact first-order information is not necessarily computable. So far we have

- established the best-known communication complexities and not improvable sampling complexities for decentralized (stochastic) optimization by proposing a class of synchronous primal-dual type communication-efficient methods. Preliminary numerical experiments on decentralized SVM models have been conducted to demon-

strate the advantages of the proposed methods comparing to some existing state-of-the-art decentralized methods.

- proposed an asynchronous decentralized algorithmic framework to maintain the established communication complexities and sampling complexities for solving a more general class of decentralized stochastic problems, for example, composite objective function given as the summation of smooth and nonsmooth convex functions. Preliminary numerical experiments had also been conducted under a simulated parallel computer environment to demonstrate the advantages of the proposed asynchronous methods.

Several interesting and worth-investigating future research directions are listed below:

- investigate privacy preserved first-order algorithms for solving machine learning problems, especially distributed machine learning problems. More specifically, incorporating rigorous methods and techniques from security and information system to design and analyze algorithms that protect data privacy under the distributed setting.

- build up rigorous distributed system and conduct more numerical experiments and computational studies for decentralized first-order methods and randomized methods for federated learning. It would also be interesting to propose communication-efficient methods to adaptively estimate problem constants, such as, the norm of Laplacian matrix $\mathcal{L}$ and maximum degree of the graph, etc.

- apply designed efficient first-order methods to solve real-world problems that have arisen from other areas, for example, manufacturing industry, data analytics in recommendation systems, signal processing and health-care, and so on.

- extend effective first-order methods and optimization techniques designed for convex optimization to non-convex optimization problems rising from deep learning, data analysis, and statistical inference, etc.

Appendices

APPENDIX A

SOME TECHNICAL PROOFS

In this chapter, we provide proofs for some important technical results, which are very helpful in the convergence analysis of the algorithms proposed in Chapter 4 and 5.

The following lemma below characterizes the solution of the primal and dual projection steps in DCS/SDCS, ADPD and AA-SDCS, as well as the projection in inner loops CS and ACS. The proof of this result can be found in Lemma 2 of [112].

Lemma A.0.1 *Let the convex function $q : U \to \mathbb{R}$, the points $\bar{x}, \bar{y} \in U$ and the scalars $\mu_1, \mu_2 \in \mathbb{R}$ be given. Let $\omega : U \to \mathbb{R}$ be a differentiable convex function and $V(x, z)$ be defined in* (4.2.7)*. If*

$$u^* \in \operatorname{argmin} \{q(u) + \mu_1 V(\bar{x}, u) + \mu_2 V(\bar{y}, u) : u \in U\},$$

then for any $u \in U$, we have

$$q(u^*) + \mu_1 V(\bar{x}, u^*) + \mu_2 V(\bar{y}, u^*) \leq q(u) + \mu_1 V(\bar{x}, u) + \mu_2 V(\bar{y}, u) - (\mu_1 + \mu_2) V(u^*, u).$$

We also define some auxiliary notations which play important roles in the convergence analysis of the proposed methods in Chapter 5. Let $\hat{\mathbf{x}}^k$, $\hat{\mathbf{y}}^k$, $\hat{\mathbf{x}}_+^k$ and $\underline{\hat{\mathbf{x}}}^k$ be defined as follows, $\forall t = 1, \ldots, k$

$$\hat{\mathbf{x}}^k = \operatorname{argmin}_{\mathbf{x} \in X^m} \langle \mathbf{L}\tilde{\mathbf{y}}^k, \mathbf{x} \rangle + F(\mathbf{x}) + \eta_t \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}), \tag{A.0.1}$$

$$\hat{\mathbf{y}}^k = \mathbf{y}^{k-1} + \tfrac{1}{\tau_k} \mathbf{L}\tilde{\mathbf{x}}^k, \tag{A.0.2}$$

$$(\hat{\mathbf{x}}_+^k, \underline{\hat{\mathbf{x}}}^k) = \mathrm{ACS}(F, X^m, \mathbf{V}, T_k, \eta_k, \mathbf{L}\tilde{\mathbf{y}}^k, \mathbf{x}^{k-1}), \tag{A.0.3}$$

and $\hat{\mathbf{x}}^0 = \hat{\mathbf{x}}_+^k = \hat{\underline{\mathbf{x}}}^0 = \mathbf{x}^0, \hat{\mathbf{y}}^0 = \mathbf{y}^0 = \mathbf{0}$. Note that some notations may be abused in the above definitions, since $\mathbf{x}^k, \tilde{\mathbf{y}}^k, \mathbf{y}^k, \tilde{\mathbf{x}}^k$ can be generated by both Algorithm 10 and Algorithm 11. However, these definitions become clear when we refer to them in the convergence analysis of certain algorithm. For example, when we refer to $\hat{\mathbf{x}}^k$ in the convergence analysis of Algorithm 10, notations $\tilde{\mathbf{y}}^k$ and $\mathbf{x}^{k-1}$ in its definition clearly refer to (5.2.6) and (5.2.8) in Algorithm 10.

In the following lemma, we provide some important relations that will be used later in the convergence analysis.

Lemma A.0.2 *For weight sequence* $\{\theta_k\}$ *defined as in* (5.2.9) *and any* $\mathbf{x} \in X^m, \mathbf{y} \in \mathbb{R}^{md}$*, we have*

$$\begin{aligned}
\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^N \theta_k[F(\mathbf{x}^k) - F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}^k, \mathbf{y}\rangle]\right\} &= \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k[F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\mathbf{x}}^k, \mathbf{y}\rangle]\right\},\\
\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^N \theta_k[F(\underline{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\underline{\mathbf{x}}^k, \mathbf{y}\rangle]\right\} &= \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^N \hat{\theta}_k[F(\hat{\underline{\mathbf{x}}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\underline{\mathbf{x}}}^k, \mathbf{y}\rangle]\right\},\\
\mathbb{E}_{[i_k,j_k]}\{\mathbf{V}(\hat{\mathbf{x}}^k, \mathbf{x})\} &= \mathbb{E}_{[i_k,j_k]}\{m\mathbf{V}(\mathbf{x}^k, \mathbf{x}) - (m-1)\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x})\},\\
\mathbb{E}_{[i_k,j_k]}\{\mathbf{V}(\hat{\mathbf{x}}^k, \mathbf{x}^{k-1})\} &= \mathbb{E}_{[i_k,j_k]}\{mV_{j_k}(x_{j_k}^k, x_{j_k}^{k-1})\},
\end{aligned}$$

where $\mathbb{E}_{[i_k,j_k]}$ *represents taking expectation over* $i_1, j_1, \ldots, i_k, j_k$*, and* $\mathbf{x}^k$*,* $\hat{\mathbf{x}}^k$*,* $\underline{\mathbf{x}}^k$ *and* $\hat{\underline{\mathbf{x}}}^k$ *are defined in* (5.2.8) *,* (A.0.1)*,* (5.2.21)*,* (A.0.3) *respectively.*

Proof. Note that by (A.0.1) and the fact that j_k is chosen uniformly from $\{1, \ldots, m\}$, we have

$$\begin{aligned}
&\mathbb{E}_{j_k}\{F(\mathbf{x}^k) - F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}^k, \mathbf{y}\rangle\}\\
&= (1 - \tfrac{1}{m})[F(\mathbf{x}^{k-1}) - F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}^{k-1}, \mathbf{y}\rangle] + \tfrac{1}{m}[F(\hat{\mathbf{x}}^{k-1}) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\mathbf{x}}^{k-1}, \mathbf{y}\rangle].
\end{aligned} \tag{A.0.4}$$

Therefore, by (5.2.9) we obtain

$$\begin{aligned}
&\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^{N}\theta_k[F(\mathbf{x}^k)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^k,\mathbf{y}\rangle]\right\}\\
&\quad=\mathbb{E}_{[i_k]}\mathbb{E}_{[j_k]}\{(\hat\theta_0-(m-1)\hat\theta_1)[F(\mathbf{x}^0)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^0,\mathbf{y}\rangle]|[i_k]\}\\
&\qquad+\mathbb{E}_{[i_k]}\mathbb{E}_{[j_k]}\left\{\textstyle\sum_{k=1}^{N-1}(m\hat\theta_k-(m-1)\hat\theta_{k+1})[F(\mathbf{x}^k)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^k,\mathbf{y}\rangle|[i_k]]\right\}\\
&\qquad+\mathbb{E}_{[i_k]}\mathbb{E}_{[j_k]}\left\{m\hat\theta_N[F(\mathbf{x}^N)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^N,\mathbf{y}\rangle]|[i_k]\right\}\\
&\quad=\mathbb{E}_{[i_k,j_k]}\left\{\hat\theta_0[F(\hat{\mathbf{x}}^0)-F(\mathbf{x})+\langle\mathbf{L}\hat{\mathbf{x}}^0,\mathbf{y}\rangle]\right\}\\
&\qquad+\mathbb{E}_{[i_k]}\mathbb{E}_{[j_k]}\left\{\textstyle\sum_{k=1}^{N}m\hat\theta_k[F(\mathbf{x}^k)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^k,\mathbf{y}\rangle]|[i_k]\right\}\\
&\qquad-\mathbb{E}_{[i_k]}\mathbb{E}_{[j_k]}\left\{\textstyle\sum_{k=1}^{N}(m-1)\hat\theta_k[F(\mathbf{x}^{k-1})-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^{k-1},\mathbf{y}\rangle]|[i_k]\right\}\\
&\quad=\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^{N}\hat\theta_k[F(\hat{\mathbf{x}}^k)-F(\mathbf{x})+\langle\mathbf{L}\hat{\mathbf{x}}^k,\mathbf{y}\rangle]\right\},
\end{aligned}$$

where the last equality is obtained by applying (A.0.4) and rearranging the terms. Similarly, in view of (A.0.3), we have

$$\begin{aligned}
&\mathbb{E}_{j_k}\{F(\underline{\mathbf{x}}^k)-F(\mathbf{x})+\langle\mathbf{L}\underline{\mathbf{x}}^k,\mathbf{y}\rangle\}\\
&=(1-\tfrac{1}{m})[F(\underline{\mathbf{x}}^{k-1})-F(\mathbf{x})+\langle\mathbf{L}\underline{\mathbf{x}}^{k-1},\mathbf{y}\rangle]+\tfrac{1}{m}[F(\underline{\hat{\mathbf{x}}}^{k-1})-F(\mathbf{x})+\langle\mathbf{L}\underline{\hat{\mathbf{x}}}^{k-1},\mathbf{y}\rangle],
\end{aligned}$$

and hence the second identity follows from the same argument. Moreover, for any $\mathbf{x}\in X^m$, $k\geq 1$, we have

$$\begin{aligned}
\mathbb{E}_{[i_k,j_k]}\{\mathbf{V}(\mathbf{x}^k,\mathbf{x})\}&=\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{j=1}^{m}V_j(x_j^k,x_j)\right\}\\
&=\mathbb{E}_{[i_k,j_k]}\left\{\tfrac{1}{m}\mathbf{V}(\hat{\mathbf{x}}^k,\mathbf{x})+(1-\tfrac{1}{m})\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})\right\},
\end{aligned}$$

where the first equality follows from the definition of $\mathbf{V}(\cdot,\cdot)$, and the last equality follows by taking expectation on j_k. Similarly, we can obtain the last relation of the lemma. ∎

We define $\hat{\mathbf{y}}^k$ (see (A.0.2)) and $\hat{\mathbf{x}}_+^k$ (see (A.0.3)) in a similar way as $\hat{\mathbf{x}}^k$, and hence, following

the same technique as in the above lemma, we can conclude

$$\begin{aligned}
\mathbb{E}_{[i_k,j_k]}\{\tilde{\mathbf{y}}^k\} &= \mathbb{E}_{[i_k,j_k]}\{\hat{\mathbf{y}}^k\},\\
\mathbb{E}_{[i_k,j_k]}\{\|\mathbf{y}-\hat{\mathbf{y}}^k\|^2\} &= \mathbb{E}_{[i_k,j_k]}\{m\|\mathbf{y}-\mathbf{y}^k\|^2-(m-1)\|\mathbf{y}-\mathbf{y}^{k-1}\|^2\},\\
\mathbb{E}_{[i_k,j_k]}\{\|\mathbf{y}^{k-1}-\hat{\mathbf{y}}^k\|^2\} &= \mathbb{E}_{[i_k,j_k]}\{m\|y_{i_k}^{k-1}-y_{i_k}^k\|^2\},
\end{aligned}$$

and

$$\mathbb{E}_{[i_k,j_k]}\{\mathbf{V}(\hat{\mathbf{x}}_+^k,\mathbf{x})\} = \mathbb{E}_{[i_k,j_k]}\{m\mathbf{V}(\mathbf{x}^k,\mathbf{x})-(m-1)\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})\},$$

where $\mathbf{x}^k$ (cf. (5.2.21)) is generated by Algorithm 11.

We now provide an important recursion relation of Algorithm 10 in the following lemma.

Lemma A.0.3 *Let the gap function Q be defined as in* (4.2.2)*, and $\bar{\mathbf{z}}^N := (\bar{\mathbf{x}}^N,\bar{\mathbf{y}}^N) = \sum_{k=0}^N(\theta_k\mathbf{x}^k,\hat{\theta}_k\hat{\mathbf{y}}^k)$, where $\{\theta_k\}$ is a nonnegative sequence that satisfies* (5.2.9)*. Also let $\mathbf{x}^k$ and $\mathbf{y}^k$ be defined in* (5.2.8) *and* (5.2.5)*, respectively. Then for any $k\geq 1$, we have*

$$\begin{aligned}
\mathbb{E}_{[i_k,j_k]}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \leq\ & \hat{\theta}_0Q_0(\mathbf{x},\mathbf{y}) + \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N\hat{\theta}_k\langle\mathbf{L}(m\mathbf{x}^k-(m-1)\mathbf{x}^{k-1}-\tilde{\mathbf{x}}^k),\mathbf{y}-\tilde{\mathbf{y}}^k\rangle\right\}\\
&+\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N m\hat{\theta}_k\eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})-\mathbf{V}(\mathbf{x}^k,\mathbf{x})-V_{j_k}(x_{j_k}^{k-1},x_{j_k}^k)]\right\}\\
&+\mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N\frac{m\hat{\theta}_k\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^k\|^2-\|y_{i_k}^{k-1}-y_{i_k}^k\|^2]\right\},
\end{aligned} \tag{A.0.5}$$

where $Q_0(\mathbf{x},\mathbf{y})$ is defined as

$$Q_0(\mathbf{x},\mathbf{y}) := F(\mathbf{x}^0)-F(\mathbf{x})+\langle\mathbf{L}\mathbf{x}^0,\mathbf{y}\rangle. \tag{A.0.6}$$

Proof. By the definitions of $Q(\cdot;\cdot)$ in (4.2.2) and $\bar{\mathbf{z}}^N$, we have

$$\begin{aligned} Q(\bar{\mathbf{z}}^N;\mathbf{z}) &= F(\bar{\mathbf{x}}^N) - F(\mathbf{x}) + \langle \mathbf{L}\bar{\mathbf{x}}^N, \mathbf{y}\rangle - \langle \mathbf{L}\mathbf{x}, \bar{\mathbf{y}}^N\rangle \\ &\le \textstyle\sum_{k=0}^N \theta_k[F(\mathbf{x}^k) - F(\mathbf{x}) + \langle \mathbf{L}\mathbf{x}^k, \mathbf{y}\rangle] - \sum_{k=0}^N \hat{\theta}_k \langle \mathbf{L}\mathbf{x}, \hat{\mathbf{y}}^k\rangle, \end{aligned}$$

where the inequality follows from the convexity of $F(\cdot)$. By taking expectation over $i_1, j_1, \ldots, i_k, j_k$ and applying Lemma A.0.2, we obtain

$$\mathbb{E}_{[i_k,j_k]}\left\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\right\} \le \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^N \hat{\theta}_k[F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\hat{\mathbf{x}}^k, \mathbf{y}\rangle - \langle \mathbf{L}\mathbf{x}, \hat{\mathbf{y}}^k\rangle]\right\}.$$

Note that by applying Lemma A.0.1 to (A.0.1) and (A.0.2), we have

$$\begin{aligned} \langle \mathbf{L}\tilde{\mathbf{y}}^k, \hat{\mathbf{x}}^k - \mathbf{x}\rangle + F(\hat{\mathbf{x}}^k) - F(\mathbf{x}) &\le \eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\hat{\mathbf{x}}^k,\mathbf{x}) - \mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)], \quad \text{(A.0.7)} \\ \langle \mathbf{L}\tilde{\mathbf{x}}^k, \mathbf{y} - \hat{\mathbf{y}}^k\rangle &\le \tfrac{\tau_k}{2}[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \hat{\mathbf{y}}^k\|^2 - \|\mathbf{y}^{k-1} - \hat{\mathbf{y}}^k\|^2]. \end{aligned}$$

Combining the above three inequalities and in view of $\hat{\mathbf{y}}^0 = \mathbf{y}^0 = \mathbf{0}$, we can conclude that

$$\begin{aligned} \mathbb{E}_{[i_k,j_k]}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \le{}& \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) + \mathbb{E}_{[i_k,j_k]}\left\{\sum_{k=1}^N \hat{\theta}_k\left[\langle \mathbf{L}\tilde{\mathbf{y}}^k, \mathbf{x} - \hat{\mathbf{x}}^k\rangle + \langle \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y}\rangle\right.\right. \\ &\left.\left. + \langle \mathbf{L}(\tilde{\mathbf{x}}^k - \mathbf{x}), \hat{\mathbf{y}}^k\rangle\right]\right\} \\ &+ \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k\eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\hat{\mathbf{x}}^k,\mathbf{x}) - \mathbf{V}(\mathbf{x}^{k-1},\hat{\mathbf{x}}^k)]\right\} \\ &+ \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N \frac{\hat{\theta}_k\tau_k}{2}[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \hat{\mathbf{y}}^k\|^2 - \|\mathbf{y}^{k-1} - \hat{\mathbf{y}}^k\|^2]\right\} \\ \le{}& \hat{\theta}_0 Q_0 + \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k\langle \mathbf{L}(\hat{\mathbf{x}}^k - \tilde{\mathbf{x}}^k), \mathbf{y} - \tilde{\mathbf{y}}^k\rangle\right\} \\ &+ \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N m\hat{\theta}_k\eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\mathbf{x}^k,\mathbf{x}) - V_{j_k}(x_{j_k}^{k-1}, x_{j_k}^k)]\right\} \\ &+ \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=1}^N \frac{m\hat{\theta}_k\tau_k}{2}[\|\mathbf{y} - \mathbf{y}^{k-1}\|^2 - \|\mathbf{y} - \mathbf{y}^k\|^2 - \|y_{i_k}^{k-1} - y_{i_k}^k\|^2]\right\}, \end{aligned} \quad \text{(A.0.8)}$$

where the second inequality follows from Lemma A.0.2 and the result in (A.0.5) immedi-

ately follows from taking expectation on j_k. ■

The following proposition establishes the main convergence property of the ADPD method stated in Algorithm 10.

Proposition A.0.4 *Let the iterates $(\mathbf{x}^k, \hat{\mathbf{y}}^k)$, $k = 1, \ldots, N$, be generated by Algorithm 10 and be defined as in* (A.0.2)*, respectively, and let $\bar{\mathbf{z}}^N := (\sum_{k=0}^N \theta_k \mathbf{x}^k, \sum_{k=0}^N \hat{\theta}_k \hat{\mathbf{y}}^k)$. Assume that the parameters $\{\alpha_k\}$, $\{\tau_k\}$, and $\{\eta_k\}$ in Algorithm 10 satisfy*

$$\hat{\theta}_k \tau_k = \hat{\theta}_{k-1} \tau_{k-1}, \quad k = 2, \ldots, N, \tag{A.0.9}$$

$$\hat{\theta}_k \eta_k \leq \hat{\theta}_{k-1} \eta_{k-1}, \quad k = 2, \ldots, N, \tag{A.0.10}$$

$$\alpha_k \hat{\theta}_k = m \hat{\theta}_{k-1}, \quad k = 2, \ldots, N+1, \tag{A.0.11}$$

$$4m\alpha_k d_{max}^2 \leq \eta_{k-1} \tau_k, \quad k = 2, \ldots, N, \tag{A.0.12}$$

$$4(m-1)^2 d_{max}^2 \leq \eta_k \tau_k, \quad k = 1, \ldots, N. \tag{A.0.13}$$

where $\{\hat{\theta}_k\}$ is some given weight sequence and d_{max} is the maximum degree of graph $\mathcal{G}$. Then, for any $\mathbf{z} := (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$, we have

$$\mathbb{E}_{[i_k, j_k]}\{Q(\bar{\mathbf{z}}^N; \mathbf{z})\} \leq \hat{\theta}_0 (F(\mathbf{x}^0) - F(\mathbf{x})) + m\hat{\theta}_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \mathbb{E}_{[i_k, j_k]} \langle \mathbf{s}, \mathbf{y} \rangle, \tag{A.0.14}$$

where Q is defined in (4.2.2)*, and $\mathbf{s}$ is defined as*

$$\mathbf{s} := \hat{\theta}_0 \mathbf{L} \mathbf{x}^0 + m\hat{\theta}_N \mathbf{L}(\mathbf{x}^N - \mathbf{x}^{N-1}) + m\hat{\theta}_1 \tau_1 \mathbf{y}^N. \tag{A.0.15}$$

Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of (4.2.1)*, we have*

$$\begin{aligned} \tfrac{m\hat{\theta}_N}{2} \left(1 - \tfrac{4d_{max}^2}{\eta_N \tau_N}\right) \max\left\{ \tfrac{\eta_N}{2} \mathbb{E}_{[i_k, j_k]} \|x_{j_N}^N - x_{j_N}^{N-1}\|^2, \tau_N \mathbb{E}_{[i_k, j_k]} \|\mathbf{y}^* - \mathbf{y}^N\|^2 \right\} \\ \leq \hat{\theta}_0 (F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^* \rangle) + m\hat{\theta}_1 \eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{m\hat{\theta}_1 \tau_1}{2} \|\mathbf{y}^*\|^2. \end{aligned} \tag{A.0.16}$$

Proof. In view of Lemma A.0.3, we have

$$\mathbb{E}_{[i_k,j_k]}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \le \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) + \mathbb{E}_{[i_k,j_k]}\left\{\sum_{k=1}^N \hat{\theta}_k \Delta_k\right\}, \tag{A.0.17}$$

where

$$\begin{aligned}\Delta_k :=& \langle \mathbf{L}(m\mathbf{x}^k - (m-1)\mathbf{x}^{k-1} - \tilde{\mathbf{x}}^k), \mathbf{y} - \tilde{\mathbf{y}}^k\rangle + m\eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\mathbf{x}^k,\mathbf{x}) - V_{j_k}(x_{j_k}^{k-1}, x_{j_k}^k)] \\ &+ \tfrac{m\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2 - \|\mathbf{y}-\mathbf{y}^k\|^2 - \|y_{i_k}^{k-1} - y_{i_k}^k\|^2].\end{aligned} \tag{A.0.18}$$

Now we will provide a bound for $\sum_{k=1}^N \hat{\theta}_k \Delta_k$. Observe that by (5.2.3), we obtain

$$\begin{aligned}&\textstyle\sum_{k=1}^N \hat{\theta}_k \langle \mathbf{L}(m\mathbf{x}^k - (m-1)\mathbf{x}^{k-1} - \tilde{\mathbf{x}}^k), \mathbf{y} - \tilde{\mathbf{y}}^k\rangle \\ &\quad= \textstyle\sum_{k=1}^N \hat{\theta}_k \langle \mathbf{L}(m(\mathbf{x}^k - \mathbf{x}^{k-1}) - \alpha_k(\mathbf{x}^{k-1} - \mathbf{x}^{k-2})), \mathbf{y} - \tilde{\mathbf{y}}^k\rangle \\ &\quad= \textstyle\sum_{k=1}^N \left[m\hat{\theta}_k \langle \mathbf{L}(\mathbf{x}^k - \mathbf{x}^{k-1}), \mathbf{y} - \tilde{\mathbf{y}}^k\rangle - \hat{\theta}_k\alpha_k \langle \mathbf{L}(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y} - \tilde{\mathbf{y}}^{k-1}\rangle\right] \\ &\qquad+ \textstyle\sum_{k=1}^N \hat{\theta}_k\alpha_k \langle \mathbf{L}(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}), \tilde{\mathbf{y}}^k - \tilde{\mathbf{y}}^{k-1})\rangle \\ &\quad= m\hat{\theta}_N \langle \mathbf{L}(\mathbf{x}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N - (m-1)(\mathbf{y}^N - \mathbf{y}^{N-1})\rangle \\ &\qquad+ \textstyle\sum_{k=2}^N \hat{\theta}_k\alpha_k \langle \mathbf{L}(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}), m(\mathbf{y}^k - \mathbf{y}^{k-1}) - (m-1)(\mathbf{y}^{k-1} - \mathbf{y}^{k-2})\rangle \\ &\quad= m\hat{\theta}_N \langle \mathbf{L}(\mathbf{x}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N\rangle + \textstyle\sum_{k=2}^N m\hat{\theta}_k\alpha_k \langle \mathbf{L}(\mathbf{x}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle \\ &\qquad- \textstyle\sum_{k=1}^N (m-1)\hat{\theta}_{k+1}\alpha_{k+1} \langle \mathbf{L}(\mathbf{x}^k - \mathbf{x}^{k-1}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle,\end{aligned}$$

where the third equality follows from (A.0.11), (5.2.6) and the fact that $\mathbf{x}^{-1} = \mathbf{x}^0$, and the last equality follows from (A.0.11) and rearranging the terms. Also note that

$$\begin{aligned}&\textstyle\sum_{k=1}^N m\hat{\theta}_k\eta_k[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\mathbf{x}^k,\mathbf{x}) - V_{j_k}(x_{j_k}^{k-1}, x_{j_k}^k)] \\ &\quad= m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}) + \textstyle\sum_{k=2}^N (m\hat{\theta}_k\eta_k - m\hat{\theta}_{k-1}\eta_{k-1})\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - m\hat{\theta}_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x}) \\ &\qquad- \textstyle\sum_{k=1}^N m\hat{\theta}_k\eta_k V_{j_k}(x_{j_k}^{k-1}, x_{j_k}^k) \\ &\quad\le m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}) - m\hat{\theta}_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x}) - \textstyle\sum_{k=1}^N m\hat{\theta}_k\eta_k V_{j_k}(x_{j_k}^{k-1}, x_{j_k}^k),\end{aligned}$$

where the last inequality follows from (A.0.10). Similarly, by (A.0.9) we have

$$\begin{aligned}&\textstyle\sum_{k=1}^{N}\frac{m\hat{\theta}_k\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2-\|\mathbf{y}-\mathbf{y}^{k}\|^2-\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2]\\&\qquad\le\tfrac{m\hat{\theta}_1\tau_1}{2}\{\|\mathbf{y}-\mathbf{y}^{0}\|^2-\|\mathbf{y}-\mathbf{y}^{N}\|^2\}-\textstyle\sum_{k=1}^{N}\frac{m\hat{\theta}_k\tau_k}{2}\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2.\end{aligned}$$

Combining the above three results, we conclude that

$$\begin{aligned}\textstyle\sum_{k=1}^{N}\hat{\theta}_k\Delta_k&\le m\hat{\theta}_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle+\textstyle\sum_{k=2}^{N}m\hat{\theta}_k\alpha_k\langle\mathbf{L}(\mathbf{x}^{k-1}-\mathbf{x}^{k-2}),\mathbf{y}^k-\mathbf{y}^{k-1}\rangle\\&\quad-\textstyle\sum_{k=1}^{N}(m-1)\hat{\theta}_{k+1}\alpha_{k+1}\langle\mathbf{L}(\mathbf{x}^{k}-\mathbf{x}^{k-1}),\mathbf{y}^k-\mathbf{y}^{k-1}\rangle\\&\quad+m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})-m\hat{\theta}_N\eta_N\mathbf{V}(\mathbf{x}^N,\mathbf{x})-\textstyle\sum_{k=1}^{N}m\hat{\theta}_k\eta_kV_{j_k}(x_{j_k}^{k-1},x_{j_k}^{k})\\&\quad+\tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}-\mathbf{y}^0\|^2-\tfrac{m\hat{\theta}_N\tau_N}{2}\|\mathbf{y}-\mathbf{y}^N\|^2-\textstyle\sum_{k=1}^{N}\frac{m\hat{\theta}_k\tau_k}{2}\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2\\&\le m\hat{\theta}_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle-\tfrac{m\hat{\theta}_N\eta_N}{4}\|x_{j_N}^{N-1}-x_{j_N}^{N}\|^2\\&\quad+\sum_{k=2}^{N}\left\{m\hat{\theta}_k\alpha_kL_{i_k,j_{k-1}}\langle x_{j_{k-1}}^{k-1}-x_{j_{k-1}}^{k-2},y_{i_k}^k-y_{i_k}^{k-1}\rangle-\tfrac{m\hat{\theta}_{k-1}\eta_{k-1}}{4}\|x_{j_{k-1}}^{k-1}-x_{j_{k-1}}^{k-2}\|^2\right\}\\&\quad+\sum_{k=1}^{N}\left\{(m-1)\hat{\theta}_{k+1}\alpha_{k+1}L_{i_k,j_k}\langle x_{j_k}^{k}-x_{j_k}^{k-1},y_{i_k}^k-y_{i_k}^{k-1}\rangle-\tfrac{m\hat{\theta}_k\eta_k}{4}\|x_{j_k}^{k-1}-x_{j_k}^{k}\|^2\right\}\\&\quad+m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\tfrac{m\hat{\theta}_1\tau_1}{2}\{\|\mathbf{y}-\mathbf{y}^0\|^2-\|\mathbf{y}-\mathbf{y}^N\|^2\}-\textstyle\sum_{k=1}^{N}\frac{m\hat{\theta}_k\tau_k}{2}\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2.\end{aligned}$$

Note that by (A.0.12) and the fact that $b\langle u,v\rangle-a\|v\|^2/2\le b^2\|u\|^2/(2a),\forall a>0$, for all $k\ge 2$, we have

$$\begin{aligned}&m\hat{\theta}_k\alpha_kL_{i_k,j_{k-1}}\langle x_{j_{k-1}}^{k-1}-x_{j_{k-1}}^{k-2},y_{i_k}^k-y_{i_k}^{k-1}\rangle-\tfrac{m\hat{\theta}_{k-1}\eta_{k-1}}{4}\|x_{j_{k-1}}^{k-1}-x_{j_{k-1}}^{k-2}\|^2-\tfrac{m\hat{\theta}_k\tau_k}{4}\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2\\&\le m\left(\tfrac{\hat{\theta}_k^2\alpha_k^2L_{i_k,j_{k-1}}^2}{\hat{\theta}_{k-1}\eta_{k-1}}-\tfrac{\hat{\theta}_k\tau_k}{4}\right)\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2\le 0.\end{aligned}$$

Similarly, by (A.0.13) for all $k\ge 1$, we have

$$(m-1)\hat{\theta}_{k+1}\alpha_{k+1}L_{i_k,i_k}\langle x_{j_k}^{k}-x_{j_k}^{k-1},y_{i_k}^k-y_{i_k}^{k-1}\rangle-\tfrac{m\hat{\theta}_k\eta_k}{4}\|x_{j_k}^{k-1}-x_{j_k}^{k}\|^2-\tfrac{m\hat{\theta}_k\tau_k}{4}\|y_{i_k}^{k-1}-y_{i_k}^{k}\|^2\le 0.$$

Hence, combining the above three inequalities, we conclude that

$$\begin{aligned}\sum_{k=1}^{N}\hat{\theta}_k\Delta_k &\le m\hat{\theta}_N\langle \mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}-\mathbf{y}^N\rangle - \tfrac{m\hat{\theta}_N\eta_N}{4}\|x_{j_N}^{N-1}-x_{j_N}^N\|^2\\ &\quad + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}) + \tfrac{m\hat{\theta}_1\tau_1}{2}\left\{\|\mathbf{y}-\mathbf{y}^0\|^2-\|\mathbf{y}-\mathbf{y}^N\|^2\right\} \qquad \text{(A.0.19)}\\ &\le m\hat{\theta}_N\langle \mathbf{L}(\mathbf{x}^{N-1}-\mathbf{x}^N),\mathbf{y}^N\rangle - \tfrac{m\hat{\theta}_N\eta_N}{4}\|x_{j_N}^{N-1}-x_{j_N}^N\|^2 - \tfrac{m\hat{\theta}_N\tau_N}{2}\|\mathbf{y}^N\|^2\\ &\quad + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}) + \tfrac{\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^0\|^2 + m\langle\hat{\theta}_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\hat{\theta}_1\tau_1(\mathbf{y}^N-\mathbf{y}^0),\mathbf{y}\rangle\\ &\le m\hat{\theta}_N\textstyle\sum_{i=1}^{m}\left(\frac{L_{i,j_N}^2}{\eta_N}-\frac{\tau_N}{2}\right)\|y_i^N\|^2 + m\langle\hat{\theta}_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\hat{\theta}_1\tau_1\mathbf{y}^N,\mathbf{y}\rangle\\ &\quad + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}),\end{aligned}$$

where the second inequality follows from (4.2.8) and the fact that $b\langle u,v\rangle - a\|v\|^2/2 \le b^2\|u\|^2/(2a), \forall a>0$, and the last inequality also follows from the fact and $\mathbf{y}^0=\mathbf{0}$. In view of (A.0.13) and (A.0.17), we obtain

$$\begin{aligned}\mathbb{E}_{[i_k,j_k]}\left\{Q(\bar{\mathbf{z}}^N,\mathbf{z})\right\} &\le \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})\\ &\quad + \mathbb{E}_{[i_k,j_k]}\left\{m\langle\hat{\theta}_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+\hat{\theta}_1\tau_1\mathbf{y}^N,\mathbf{y}\rangle\right\}\\ &= \hat{\theta}_0(F(\mathbf{x}^0)-F(\mathbf{x})) + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x})\\ &\quad + \mathbb{E}_{[i_k,j_k]}\left\{\langle\hat{\theta}_0\mathbf{L}\mathbf{x}^0 + m\hat{\theta}_N\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1})+m\hat{\theta}_1\tau_1\mathbf{y}^N,\mathbf{y}\rangle\right\},\end{aligned} \qquad \text{(A.0.20)}$$

where the last equality follows from the definition of Q_0 in (A.0.6). The result in (A.0.14) immediately follows from the above relation. Furthermore, from (A.0.17), (A.0.19), (A.0.9) and the facts that $Q(\bar{\mathbf{z}}^N,\mathbf{z}^*)\ge 0, \mathbf{y}^0=\mathbf{0}$, we have

$$\begin{aligned}0\le\mathbb{E}_{[i_k,j_k]}\left\{Q(\bar{\mathbf{z}}^N,\mathbf{z}^*)\right\} &\le \hat{\theta}_0 Q_0(\mathbf{x}^*,\mathbf{y}^*) + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*) + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2\\ &\quad + \mathbb{E}_{[i_k,j_k]}\left\{m\hat{\theta}_N\langle\mathbf{L}(\mathbf{x}^N-\mathbf{x}^{N-1}),\mathbf{y}^*-\mathbf{y}^N\rangle - \tfrac{m\hat{\theta}_N\eta_N}{4}\|x_{j_N}^{N-1}-x_{j_N}^N\|^2\right\}\\ &\quad - \mathbb{E}_{[i_k,j_k]}\tfrac{m\hat{\theta}_N\tau_N}{2}\left\{\|\mathbf{y}^*-\mathbf{y}^N\|^2\right\}\end{aligned}$$

$$
\begin{aligned}
&\leq \hat{\theta}_0 Q_0(\mathbf{x}^*, \mathbf{y}^*) + m\hat{\theta}_1\eta_1 \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 \\
&\quad + \mathbb{E}_{[i_k, j_k]}\left\{\textstyle\sum_{i=1}^m m\hat{\theta}_N L_{i,j_N}\langle x_{j_N}^N - x_{j_N}^{N-1}, y_i^* - y_i^N\rangle\right\} \\
&\quad - \mathbb{E}_{[i_k, j_k]}\left\{\tfrac{m\hat{\theta}_N\tau_N}{2}\textstyle\sum_{i=1}^m\|y_i^* - y_i^N\|^2 + \tfrac{m\hat{\theta}_N\eta_N}{4}\|x_{j_N}^{N-1} - x_{j_N}^N\|^2\right\},
\end{aligned}
$$

which together with (A.0.6) and the fact that $b\langle u, v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a), \forall a > 0$ imply that

$$
\begin{aligned}
\tfrac{m\hat{\theta}_N\eta_N}{4}\mathbb{E}_{[i_k, j_k]}\|x_{j_N}^{N-1} - x_{j_N}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 \\
+ \mathbb{E}_{[i_k, j_k]}\{\tfrac{m\hat{\theta}_N d_{max}^2}{\tau_N}\|x_{j_N}^{N-1} - x_{j_N}^N\|^2\},
\end{aligned}
$$

where the last inequality follows from the definition of $\mathcal{L}$ in (1.2.18). Similarly, we obtain

$$
\begin{aligned}
\tfrac{m\hat{\theta}_N\tau_N}{2}\mathbb{E}_{[i_k, j_k]}\|\mathbf{y}^* - \mathbf{y}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1\eta_1\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 \\
+ \mathbb{E}_{[i_k, j_k]}\{\tfrac{m\hat{\theta}_N d_{max}^2}{\eta_N}\|\mathbf{y}^* - \mathbf{y}^N\|^2\},
\end{aligned}
$$

which implies the result in (A.0.16). ∎

We state in the following proposition a general result for the ACS procedure. For notation convenience, we use the notations defined the in ACS procedure (cf. Algorithm 11) and let

$$
\Lambda_t := \begin{cases} 1, & t = 1, \\ (1 - \lambda_t)\Lambda_t, & t \geq 2. \end{cases} \tag{A.0.21}
$$

Proposition A.0.5 *If* $\{\beta_t\}$ *and* $\{\lambda_t\}$ *in the* ACS *procedure satisfy*

$$
\lambda_1 = 1, \tag{A.0.22}
$$

$$
\mu + \eta + \beta_t > (\mathcal{C} + L)\lambda_t^2,\ t = 1, \ldots, T, \tag{A.0.23}
$$

$$
\tfrac{\beta_t}{\Lambda_t} = \tfrac{\beta_{t-1}}{\Lambda_{t-1}},\ t = 1, \ldots, T, \tag{A.0.24}
$$

then, under assumptions (4.4.1) *and* (4.4.2)*, for* $u \in U$*,*

$$\mathbb{E}_{[\xi]}\Phi(\underline{u}^T) - \Phi(u) \leq \Lambda_T\beta_1 V(u^0,u) - (\Lambda_T\beta_1+\mu+\eta)\mathbb{E}_\xi V(u^T,u) + \Lambda_T\sum_{t=1}^T \frac{2(M^2+\sigma^2)\lambda_t^2}{(\mu+\eta+\beta_t-(\mathcal{C}+L)\lambda_t^2)\Lambda_t}, \tag{A.0.25}$$

where $\mathbb{E}_{[\xi]}$ *represents taking the expectation over* $\{\xi_i^1,\ldots,\xi_i^T\}$ *and* Φ *is defined as*

$$\Phi(u) := \langle w,u\rangle + \phi(u) + \eta V(x,u). \tag{A.0.26}$$

Proof. Note that in view of (4.2.7), (4.2.8) and (4.3.32), we have

$$\tfrac{1}{2}\|u_1-u_2\|^2 \leq V(x,u_1)-V(x,u_2)-\langle\nabla V(x,u_2),u_1-u_2\rangle = V(u_2,u_1) \leq \tfrac{\mathcal{C}}{2}\|u_1-u_2\|^2,\ \forall u_1,u_2\in U,$$

where $\nabla V(x,u_2)$ denotes the gradient of $V(x,\cdot)$ w.r.t. u_2 for a given x, and the above result together with (5.1.1) imply $\phi(\cdot)$ satisfies

$$\tfrac{\mu+\eta}{2}\|u_1-u_2\|^2 \leq \Phi(u_1)-\Phi(u_2)-\langle\nabla\Phi(u_2),u_1-u_2\rangle \leq \tfrac{\mathcal{C}+L}{2}\|u_1-u_2\|^2+M\|u_1-u_2\|,\ \forall u_1,u_2\in U.$$

Hence, by the proof of Theorem 1 in [112], we can conclude that

$$\mathbb{E}_{[\xi]}\Phi(\underline{u}^T) - \Phi(u) \leq \Lambda_T\beta_1 V(u^0,u) - (\Lambda_T\beta_1+\mu+\eta)\mathbb{E}_{[\xi]} V(u^T,u) + \Lambda_t\sum_{t=1}^T \frac{2\lambda_t^2(M^2+\sigma^2)}{\Lambda_t(\mu+\eta+\beta_t-(\mathcal{C}+L)\lambda_t^2)}.$$

■

We are now ready to present the main convergence property of the AA-SDCS method stated in Algorithm 11 when the objective functions $f_i, i=1,\ldots,m$, are general convex.

Proposition A.0.6 *Let the iterates* $(\underline{\mathbf{x}}^k,\mathbf{x}^k)$ *and* $\hat{\mathbf{y}}^k$*,* $k=1,\ldots,N$*, be generated by Algorithm 11 and be defined as in* (A.0.2)*, respectively, and let* $\bar{\mathbf{z}}^N := (\sum_{k=0}^N\theta_k\underline{\mathbf{x}}^k, \sum_{k=0}^N\hat{\theta}_k\hat{\mathbf{y}}^k)$*. Assume that the objective* $f_i, i=1,\ldots,m$*, are general convex functions, i.e.,* $\mu=0,\ L,M\geq$

0 *in* (5.1.1)*. Let the parameters* $\{\alpha_k\}$*,* $\{\tau_k\}$*, and* $\{\eta_k\}$ *in Algorithm 11 satisfy* (A.0.9) *and*

$$\hat{\theta}_k\left(\tfrac{\mathcal{C}+L}{T_k(T_k+1)}+\eta_k\right)\le\hat{\theta}_{k-1}\left(\tfrac{\mathcal{C}+L}{T_{k-1}(T_{k-1}+1)}+\eta_{k-1}\right),\ k=2,\dots,N, \tag{A.0.27}$$

$$\alpha_k\hat{\theta}_k=\hat{\theta}_{k-1},\ k=2,\dots,N, \tag{A.0.28}$$

$$8m\alpha_k d_{max}^2\le\eta_{k-1}\tau_k, k=2,\dots,N, \tag{A.0.29}$$

$$8(m-1)^2 d_{max}^2\le m\eta_k\tau_k, k=1,\dots,N, \tag{A.0.30}$$

where $\{\hat{\theta}_k\}$ *is some given weight sequence. Let the parameters* $\{\lambda_t\}$ *and* $\{\beta_t\}$ *in the* ACS *procedure of Algorithm 11 be set to* (5.2.27)*. Then, for any* $\mathbf{z}:=(\mathbf{x},\mathbf{y})\in X^m\times\mathbb{R}^{md}$*, we have*

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\}\le\hat{\theta}_0(F(\mathbf{x}^0)-F(\mathbf{x}))+m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)}+\eta_1\right)\mathbf{V}(\mathbf{x}^0,\mathbf{x})+\mathbb{E}\{\langle\mathbf{s},\mathbf{y}\rangle\}+\textstyle\sum_{k=1}^N\frac{8m(M^2+\sigma^2)\hat{\theta}_k}{\eta_k(T_k+1)}, \tag{A.0.31}$$

where $\mathbb{E}$ *represents taking the expectation over all random variables,* Q *is defined in* (4.2.2) *and* $\mathbf{s}$ *are defined as*

$$\mathbf{s}:=\hat{\theta}_0\mathbf{L}\mathbf{x}^0+\hat{\theta}_N\mathbf{L}(\underline{\hat{\mathbf{x}}}^N-\mathbf{x}^{N-1})+m\hat{\theta}_1\tau_1\mathbf{y}^N. \tag{A.0.32}$$

Furthermore, for any saddle point $(\mathbf{x}^*,\mathbf{y}^*)$ *of* (4.2.1)*, we have*

$$\begin{aligned}&\tfrac{\hat{\theta}_N}{4}\left(1-\tfrac{2\|\mathbf{L}\|^2}{m\eta_N\tau_N}\right)\max\left\{\eta_N\mathbb{E}\|\underline{\hat{\mathbf{x}}}^N-\mathbf{x}^{N-1}\|^2,2m\tau_N\mathbb{E}\|\mathbf{y}^*-\mathbf{y}^N\|^2\right\}\\&\quad\le\hat{\theta}_0(F(\mathbf{x}^0)-F(\mathbf{x}^*)+\langle\mathbf{L}\mathbf{x}^0,\mathbf{y}^*\rangle)+m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)}+\eta_1\right)\mathbf{V}(\mathbf{x}^0,\mathbf{x}^*)\\&\qquad+\tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2+\textstyle\sum_{k=1}^N\frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k}.\end{aligned} \tag{A.0.33}$$

Proof. Since f_i's are general convex function, we have $\mu=0$ and $L,M\ge0$ (cf. (5.1.1)). Also note that λ_t and β_t defined in (5.2.27) satisfy condition (A.0.22)-(A.0.24). Therefore, substituting $\phi:=f_i$, and λ_t and β_t, relation (A.0.25) can be rewritten as the

following,[1]

$$\mathbb{E}_{[\xi]}\Phi_i(\underline{u}_i^T) - \Phi_i(u_i) \leq \Lambda_T\beta_1 V_i(u_i^0, u_i) - (\Lambda_T\beta_1 + \eta)\mathbb{E}_\xi V_i(u_i^T, u_i) + \Lambda_T\sum_{t=1}^T \frac{2(M^2+\sigma^2)\lambda_t^2}{(\eta+\beta_t-(\mathcal{C}+L)\lambda_t^2)\Lambda_t},$$

Summing up the above inequality from $i \in [m]$, and using the definitions of $\hat{\mathbf{x}}_+^k$ and $\underline{\hat{\mathbf{x}}}^k$ in (A.0.3), we obtain

$$\mathbb{E}_{[\xi]}\Phi^k(\underline{\hat{\mathbf{x}}}^k) - \Phi^k(\mathbf{x}) \leq \Lambda_{T_k}\beta_1 \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - (\Lambda_{T_k}\beta_1 + \eta_k)\mathbb{E}_\xi \mathbf{V}(\mathbf{x}_+^k, \mathbf{x}) + \Lambda_{T_k}\sum_{t=1}^{T_k} \frac{2m(M^2+\sigma^2)\lambda_t^2}{(\eta_k+\beta_t-(\mathcal{C}+L)\lambda_t^2)\Lambda_t},$$

where $\Phi^k(\mathbf{x}) = \langle \mathbf{L}\mathbf{x}, \tilde{\mathbf{y}}^k\rangle + F(\mathbf{x}) + \eta_k \mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x})$. By plugging into the above relation the values of λ_t and β_t in (5.2.27), together with the definition of $\Phi^k(\mathbf{x})$ and rearranging the terms, we have $\forall \mathbf{x} \in X^m$

$$\begin{aligned}\mathbb{E}_{[\xi]}\left\{\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \mathbf{x}), \tilde{\mathbf{y}}^k\rangle + F(\underline{\hat{\mathbf{x}}}^k) - F(\mathbf{x})\right\} \leq \left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)} + \eta_k\right)\mathbb{E}_{[\xi]}\left[\mathbf{V}(\mathbf{x}^{k-1}, \mathbf{x}) - \mathbf{V}(\hat{\mathbf{x}}_+^k, \mathbf{x})\right] \\ - \eta_k\mathbb{E}_{[\xi]}\{\mathbf{V}(\mathbf{x}^{k-1}, \underline{\hat{\mathbf{x}}}^k)\} + \tfrac{8m(M^2+\sigma^2)}{(T_k+1)\eta_k}.\end{aligned} \tag{A.0.34}$$

By the definitions of Q in (4.2.2) and $\bar{\mathbf{z}}^N$, and the convexity of $F(\cdot)$, we have

$$Q(\bar{\mathbf{z}}^N; \mathbf{z}) \leq \textstyle\sum_{k=0}^N \theta_k[F(\underline{\mathbf{x}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\underline{\mathbf{x}}^k, \mathbf{y}\rangle] - \sum_{k=0}^N \hat{\theta}_k\langle \mathbf{L}\mathbf{x}, \hat{\mathbf{y}}^k\rangle.$$

Taking expectation over $i_1, j_1, \ldots, i_k, j_k$ and applying Lemma A.0.2, we obtain

$$\mathbb{E}_{[i_k,j_k]}\left\{Q(\bar{\mathbf{z}}^N; \mathbf{z})\right\} \leq \mathbb{E}_{[i_k,j_k]}\left\{\textstyle\sum_{k=0}^N \hat{\theta}_k[F(\underline{\hat{\mathbf{x}}}^k) - F(\mathbf{x}) + \langle \mathbf{L}\underline{\hat{\mathbf{x}}}^k, \mathbf{y}\rangle - \langle \mathbf{L}\mathbf{x}, \hat{\mathbf{y}}^k\rangle]\right\}.$$

Moreover, if we replace (A.0.7) by (A.0.34) in Lemma A.0.3, we can conclude the follow-

[1] We added the subscript i to emphasize that this inequality holds for any agent $i \in \mathcal{N}$ with $\phi = f_i$. More specifically, $\Phi_i(u_i) := \langle w_i, u_i\rangle + f_i(u_i) + \eta V_i(x_i, u_i)$.

ing result similar to (A.0.8)

$$\begin{aligned}\mathbb{E}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \le \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) &+ \textstyle\sum_{k=1}^N \frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k} \\ &+ \mathbb{E}\textstyle\sum_{k=1}^N \left\{\hat{\theta}_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \tilde{\mathbf{x}}^k), \mathbf{y}-\tilde{\mathbf{y}}^k\rangle - \hat{\theta}_k\eta_k \mathbf{V}(\mathbf{x}^{k-1}, \underline{\hat{\mathbf{x}}}^k)]\right\} \\ &+ \mathbb{E}\left\{\textstyle\sum_{k=1}^N m\hat{\theta}_k\left(\frac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k\right)[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\mathbf{x}^k,\mathbf{x})]\right\} \\ &+ \mathbb{E}\left\{\textstyle\sum_{k=1}^N \frac{m\hat{\theta}_k\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2 - \|\mathbf{y}-\mathbf{y}^k\|^2 - \|y_{i_k}^{k-1} - y_{i_k}^k\|^2]\right\},\end{aligned}$$

where $\mathbb{E}$ represents taking the expectation over all random variables. Therefore, we have

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \le \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) + \textstyle\sum_{k=1}^N \frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k} + \mathbb{E}\left\{\sum_{k=1}^N \hat{\theta}_k\tilde{\Delta}_k\right\}, \tag{A.0.35}$$

where

$$\begin{aligned}\tilde{\Delta}_k :=& \langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \tilde{\mathbf{x}}^k), \mathbf{y}-\tilde{\mathbf{y}}^k\rangle + m\left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k\right)[\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \mathbf{V}(\mathbf{x}^k,\mathbf{x})] \\ &- \eta_k\mathbf{V}(\mathbf{x}^{k-1},\underline{\hat{\mathbf{x}}}^k) + \tfrac{m\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2 - \|\mathbf{y}-\mathbf{y}^k\|^2 - \|y_{i_k}^{k-1} - y_{i_k}^k\|^2]. \end{aligned}\tag{A.0.36}$$

We now provide a bound for $\mathbb{E}\{\sum_{k=1}^N \hat{\theta}_k\tilde{\Delta}_k\}$. Observe that $\tilde{\Delta}_k$ is different from Δ_k defined in (A.0.18) in first three terms, however, they can be bounded via the same technique. Note that by (5.2.16), we obtain

$$\begin{aligned}&\mathbb{E}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \tilde{\mathbf{x}}^k), \mathbf{y}-\tilde{\mathbf{y}}^k\rangle\right\} \\ &= \mathbb{E}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k\langle \mathbf{L}((\underline{\hat{\mathbf{x}}}^k - \mathbf{x}^{k-1}) - \alpha_k(m\underline{\mathbf{x}}^{k-1} - (m-1)\underline{\mathbf{x}}^{k-2} - \mathbf{x}^{k-2}), \mathbf{y}-\tilde{\mathbf{y}}^k\rangle\right\} \\ &= \mathbb{E}\left\{\textstyle\sum_{k=1}^N \left[\hat{\theta}_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \mathbf{x}^{k-1}), \mathbf{y}-\tilde{\mathbf{y}}^k\rangle - \hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y}-\tilde{\mathbf{y}}^{k-1}\rangle\right]\right\} \\ &\quad + \mathbb{E}\left\{\textstyle\sum_{k=1}^N \hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), \tilde{\mathbf{y}}^k - \tilde{\mathbf{y}}^{k-1})\rangle\right\} \\ &\overset{(A.0.28),(5.2.19)}{=} \hat{\theta}_N\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y}-\mathbf{y}^N - (m-1)(\mathbf{y}^N - \mathbf{y}^{N-1})\rangle \\ &\quad + \textstyle\sum_{k=2}^N \hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), m(\mathbf{y}^k - \mathbf{y}^{k-1}) - (m-1)(\mathbf{y}^{k-1} - \mathbf{y}^{k-2})\rangle \\ &\overset{(A.0.28)}{=} \hat{\theta}_N\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y}-\mathbf{y}^N\rangle + \textstyle\sum_{k=2}^N m\hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle\end{aligned}$$

$$- \sum_{k=1}^{N}(m-1)\hat{\theta}_{k+1}\alpha_{k+1}\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \mathbf{x}^{k-1}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle,$$

which together with (A.0.27) and (A.0.9) imply that

$$\begin{aligned}
\mathbb{E}\{\textstyle\sum_{k=1}^{N}\hat{\theta}_k\tilde{\Delta}_k\} &\le \mathbb{E}\left\{\hat{\theta}_N\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N\rangle + \textstyle\sum_{k=2}^{N} m\hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle\right\} \\
&\quad - \mathbb{E}\left\{\textstyle\sum_{k=1}^{N}(m-1)\hat{\theta}_{k+1}\alpha_{k+1}\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \mathbf{x}^{k-1}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle + \textstyle\sum_{k=1}^{N}\hat{\theta}_k\eta_k\mathbf{V}(\mathbf{x}^{k-1}, \underline{\hat{\mathbf{x}}}^k)\right\} \\
&\quad + \mathbb{E}\left\{m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right)\mathbf{V}(\mathbf{x}^0, \mathbf{x}) - m\hat{\theta}_N\left(\tfrac{4(\mathcal{C}+L)}{T_N(T_N+1)} + \eta_N\right)\mathbf{V}(\mathbf{x}^N, \mathbf{x})\right\} \\
&\quad + \mathbb{E}\left\{\tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y} - \mathbf{y}^0\|^2 - \tfrac{m\hat{\theta}_N\tau_N}{2}\|\mathbf{y} - \mathbf{y}^N\|^2 - \textstyle\sum_{k=1}^{N}\tfrac{m\hat{\theta}_k\tau_k}{2}\|y_{i_k}^{k-1} - y_{i_k}^k\|^2\right\} \\
&\le \mathbb{E}\left\{\hat{\theta}_N\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N\rangle - \tfrac{\hat{\theta}_N\eta_N}{4}\|\mathbf{x}^{N-1} - \underline{\hat{\mathbf{x}}}^N\|^2 - \textstyle\sum_{k=1}^{N}\tfrac{m\hat{\theta}_k\tau_k}{2}\|y_{i_k}^{k-1} - y_{i_k}^k\|^2\right\} \\
&\quad + \textstyle\sum_{k=2}^{N}\mathbb{E}\left\{m\hat{\theta}_k\alpha_k\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle - \tfrac{\hat{\theta}_{k-1}\eta_{k-1}}{4}\|\underline{\hat{\mathbf{x}}}^{k-1} - \mathbf{x}^{k-2}\|^2\right\} \\
&\quad + \textstyle\sum_{k=1}^{N}\mathbb{E}\left\{(m-1)\hat{\theta}_{k+1}\alpha_{k+1}\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^k - \mathbf{x}^{k-1}), \mathbf{y}^k - \mathbf{y}^{k-1}\rangle - \tfrac{\hat{\theta}_k\eta_k}{4}\|\mathbf{x}^{k-1} - \underline{\hat{\mathbf{x}}}^k\|^2\right\} \\
&\quad + m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right)\mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{m\hat{\theta}_1\tau_1}{2}\mathbb{E}\{\|\mathbf{y} - \mathbf{y}^0\|^2 - \|\mathbf{y} - \mathbf{y}^N\|^2\}.
\end{aligned}$$

Noting that by the fact that $b\langle u, v\rangle - a\|v\|^2/2 \le b^2\|u\|^2/(2a), \forall a > 0$ and (A.0.28) and (A.0.29), for all $k \ge 2$, we have

$$\begin{aligned}
&\textstyle\sum_{j=1}^{m}\left\{m\hat{\theta}_k\alpha_k L_{i_k,j}\langle \underline{\hat{x}}_j^{k-1} - x_j^{k-2}, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \tfrac{\hat{\theta}_{k-1}\eta_{k-1}}{4}\|\underline{\hat{x}}_j^{k-1} - x_j^{k-2}\|^2\right\} - \tfrac{m\hat{\theta}_k\tau_k}{4}\|y_{i_k}^{k-1} - y_{i_k}^k\|^2 \\
&\le \left(\textstyle\sum_{j=1}^{m}\tfrac{m\hat{\theta}_k^2\alpha_k^2 L_{i_k,j_{k-1}}^2}{\hat{\theta}_{k-1}\eta_{k-1}} - \tfrac{\hat{\theta}_k\tau_k}{4}\right)\|y_{i_k}^{k-1} - y_{i_k}^k\|^2 \le 0.
\end{aligned}$$

Similarly, by (A.0.30) for all $k \ge 1$, we have

$$\textstyle\sum_{j=1}^{m}\left\{(m-1)\hat{\theta}_{k+1}\alpha_{k+1}L_{i_k,j}\langle \underline{\hat{x}}_j^k - x_j^{k-1}, y_{i_k}^k - y_{i_k}^{k-1}\rangle - \tfrac{\hat{\theta}_k\eta_k}{4}\|x_j^{k-1} - \underline{\hat{x}}_j^k\|^2\right\} - \tfrac{m\hat{\theta}_k\tau_k}{4}\|y_{i_k}^{k-1} - y_{i_k}^k\|^2 \le 0.$$

Hence, in view of the above three results, we obtain

$$\mathbb{E}\left\{\textstyle\sum_{k=1}^{N}\hat{\theta}_k\tilde{\Delta}_k\right\} \le \hat{\theta}_N\mathbb{E}\{\langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y} - \mathbf{y}^N\rangle\} - \tfrac{\hat{\theta}_N\eta_N}{4}\mathbb{E}\{\|\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}\|^2\}$$

$$+ m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \tfrac{m\hat{\theta}_1\tau_1}{2}\mathbb{E}\{\|\mathbf{y} - \mathbf{y}^0\|^2 - \|\mathbf{y} - \mathbf{y}^N\|^2\}. \tag{A.0.37}$$

Following the same procedure as we used in Proposition A.0.4 (cf. (A.0.20)), and using the above result and (A.0.35), we can conclude that

$$\begin{aligned}\mathbb{E}\left\{Q(\bar{\mathbf{z}}^N, \mathbf{z})\right\} \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x})) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \textstyle\sum_{k=1}^N \tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k} \\ + \mathbb{E}\left\{\langle \hat{\theta}_0 \mathbf{L}\mathbf{x}^0 + \hat{\theta}_N \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}) + m\hat{\theta}_1\tau_1 \mathbf{y}^N, \mathbf{y}\rangle\right\},\end{aligned}$$

which implies the result in (A.0.31). Furthermore, from (A.0.35), (A.0.37), (A.0.9), and the fact that $Q(\bar{\mathbf{z}}^N, \mathbf{z}^*) \geq 0, \mathbf{y}^0 = \mathbf{0}$, we have

$$\begin{aligned}0 \leq \mathbb{E}\{Q(\bar{\mathbf{z}}^N, \mathbf{z}^*)\} \leq \hat{\theta}_0 Q_0(\mathbf{x}^*, \mathbf{y}^*) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 \\ + \mathbb{E}\left\{\hat{\theta}_N \langle \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}), \mathbf{y}^* - \mathbf{y}^N\rangle - \tfrac{\hat{\theta}_N\eta_N}{4}\|\underline{\hat{\mathbf{x}}}^{N-1} - \mathbf{x}^N\|^2\right\} \\ - \mathbb{E}\left\{\tfrac{m\hat{\theta}_N\tau_N}{2}\|\mathbf{y}^* - \mathbf{y}^N\|^2\right\} + \textstyle\sum_{k=1}^N \tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k},\end{aligned}$$

which together with the fact that $b\langle u, v\rangle - a\|v\|^2/2 \leq b^2\|u\|^2/(2a), \forall a > 0$ and (A.0.6) imply that

$$\begin{aligned}\tfrac{\hat{\theta}_N\eta_N}{4}\mathbb{E}\|\underline{\hat{\mathbf{x}}}^{N-1} - \mathbf{x}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) \\ + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 + \mathbb{E}\left\{\tfrac{\hat{\theta}_N\|\mathbf{L}\|^2}{2m\tau_N}\|\underline{\hat{\mathbf{x}}}^{N-1} - \mathbf{x}^N\|^2\right\} + \textstyle\sum_{k=1}^N \tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k}.\end{aligned} \tag{A.0.38}$$

Similarly, we can obtain

$$\begin{aligned}\tfrac{m\hat{\theta}_N\tau_N}{2}\mathbb{E}\|\mathbf{y}^* - \mathbf{y}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) \\ + \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 + \mathbb{E}\{\tfrac{m\hat{\theta}_N\|\mathbf{L}\|^2}{\eta_N}\|\mathbf{y}^* - \mathbf{y}^N\|^2\} + \textstyle\sum_{k=1}^N \tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)\eta_k},\end{aligned}$$

which implies the result in (A.0.33). ■

In the following proposition, we provide the main convergence property of the AA-SDCS method stated in Algorithm 11 when the objective functions $f_i, i = 1, \ldots, m$, are strongly convex.

Proposition A.0.7 *Let the iterates $(\underline{\mathbf{x}}^k, \mathbf{x}^k)$ and $\hat{\mathbf{y}}^k$, $k = 1, \ldots, N$, be generated by Algorithm 11 and be defined as in* (A.0.2)*, respectively, and let $\bar{\mathbf{z}}^N := (\sum_{k=0}^N \theta_k \underline{\mathbf{x}}^k, \sum_{k=0}^N \hat{\theta}_k \hat{\mathbf{y}}^k)$. Assume that the objective $f_i, i = 1, \ldots, m$, are strongly convex functions, i.e., $\mu > 0$, $L, M \geq 0$ in* (5.1.1)*. Let the parameters $\{\alpha_k\}$, $\{\tau_k\}$, and $\{\eta_k\}$ in Algorithm 11 satisfy* (A.0.9)*,* (A.0.28) - (A.0.30)*,*

$$\hat{\theta}_k \left(\tfrac{\mathcal{C}+L}{T_k(T_k+1)} + \eta_k \right) \leq \hat{\theta}_{k-1} \left(\tfrac{\mathcal{C}+L}{T_{k-1}(T_{k-1}+1)} + \eta_{k-1} + \mu \right), \; k = 2, \ldots, N, \tag{A.0.39}$$

where $\{\hat{\theta}_k\}$ is some given weight sequence. Let the parameters $\{\lambda_t\}$ and $\{\beta_t\}$ in the ACS *procedure of Algorithm 11 be set to* (5.2.27)*. Then, for any $\mathbf{z} := (\mathbf{x}, \mathbf{y}) \in X^m \times \mathbb{R}^{md}$, we have*

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N; \mathbf{z})\} \leq \hat{\theta}_0 (F(\mathbf{x}^0) - F(\mathbf{x})) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1 \right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \mathbb{E}\{\langle \mathbf{s}, \mathbf{y} \rangle\} + \textstyle\sum_{k=1}^N \frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(\eta_k+\mu)(T_k+1)}, \tag{A.0.40}$$

where $\mathbb{E}$ represents the taking expectation over all random variables, Q and $\mathbf{s}$ are defined in (4.2.2) *and* (A.0.32) *respectively. Furthermore, for any saddle point $(\mathbf{x}^*, \mathbf{y}^*)$ of* (4.2.1)*, we have*

$$\begin{aligned} &\tfrac{\hat{\theta}_N}{4} \left(1 - \tfrac{2\|\mathbf{L}\|^2}{m\eta_N\tau_N} \right) \max \left\{ \eta_N \mathbb{E}\|\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}\|^2, 2m\tau_N \mathbb{E}\|\mathbf{y}^* - \mathbf{y}^N\|^2 \right\} \\ &\quad \leq \hat{\theta}_0 (F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^* \rangle) + m\hat{\theta}_1 \left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1 \right) \mathbf{V}(\mathbf{x}^0, \mathbf{x}^*) \\ &\qquad + \tfrac{m\hat{\theta}_1\tau_1}{2} \|\mathbf{y}^*\|^2 + \textstyle\sum_{k=1}^N \frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)(\eta_k+\mu)}. \end{aligned} \tag{A.0.41}$$

Proof. Since f_i's are strongly convex function, we have $\mu > 0$ and $L, M \geq 0$ (cf.

(5.1.1)). Observe that λ_t and β_t defined in (5.2.27) satisfy conditions (A.0.22)-(A.0.24). Therefore, following similar procedure in Proposition A.0.6, in view of Proposition A.0.5, and the definition of $\hat{\mathbf{x}}_+^k$ and $\underline{\hat{\mathbf{x}}}^k$ in (A.0.3), we can obtain

$$\begin{aligned}\mathbb{E}_{[\xi]}\Phi^k(\underline{\hat{\mathbf{x}}}^k) - \Phi^k(\mathbf{x}) \leq \Lambda_{T_k}\beta_1\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - (\Lambda_{T_k}\beta_1 + \mu + \eta_k)\mathbb{E}_\xi\mathbf{V}(\mathbf{x}_+^k,\mathbf{x}) \\ + \Lambda_{T_k}\textstyle\sum_{t=1}^{T_k}\frac{2m(M^2+\sigma^2)\lambda_t^2}{(\mu+\eta_k+\beta_t-(\mathcal{C}+L)\lambda_t^2)\Lambda_t},\end{aligned}$$

where $\Phi^k(\mathbf{x}) = \langle\mathbf{L}\mathbf{x},\tilde{\mathbf{y}}^k\rangle + F(\mathbf{x}) + \eta_k\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})$. By plugging into the above relation the values of λ_t and β_t in (5.2.27), together with the definition of $\Phi^k(\mathbf{x})$ and rearranging the terms, we have $\forall\mathbf{x}\in X^m$

$$\begin{aligned}\mathbb{E}_{[\xi]}\left\{\langle\mathbf{L}(\underline{\hat{\mathbf{x}}}^k-\mathbf{x}),\tilde{\mathbf{y}}^k\rangle + F(\underline{\hat{\mathbf{x}}}^k) - F(\mathbf{x})\right\} \leq \left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k\right)\mathbb{E}_{[\xi]}\{\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})\} \\ - \left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k+\mu\right)\mathbb{E}_{[\xi]}\{\mathbf{V}(\hat{\mathbf{x}}_+^k,\mathbf{x})\} - \eta_k\mathbb{E}_{[\xi]}\{\mathbf{V}(\mathbf{x}^{k-1},\underline{\hat{\mathbf{x}}}^k)\} + \tfrac{8m(M^2+\sigma^2)}{(T_k+1)(\eta_k+\mu)}.\end{aligned} \tag{A.0.42}$$

Observe that if we replace (A.0.34) by (A.0.42) in Proposition A.0.6, we can conclude the following result similar to (A.0.35)

$$\mathbb{E}\{Q(\bar{\mathbf{z}}^N;\mathbf{z})\} \leq \hat{\theta}_0 Q_0(\mathbf{x},\mathbf{y}) + \textstyle\sum_{k=1}^N\frac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)(\eta_k+\mu)} + \mathbb{E}\left\{\sum_{k=1}^N\hat{\theta}_k\bar{\Delta}_k\right\}, \tag{A.0.43}$$

where $\mathbb{E}$ represents taking the expectation over all random variables and

$$\begin{aligned}\bar{\Delta}_k :=& \langle\mathbf{L}(\underline{\hat{\mathbf{x}}}^k-\tilde{\mathbf{x}}^k),\mathbf{y}-\tilde{\mathbf{y}}^k\rangle + m\left[\left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k\right)\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x}) - \left(\tfrac{4(\mathcal{C}+L)}{T_k(T_k+1)}+\eta_k+\mu\right)\mathbf{V}(\mathbf{x}^k,\mathbf{x})\right] \\ & - \eta_k\mathbf{V}(\mathbf{x}^{k-1},\underline{\hat{\mathbf{x}}}^k) + \tfrac{m\tau_k}{2}[\|\mathbf{y}-\mathbf{y}^{k-1}\|^2 - \|\mathbf{y}-\mathbf{y}^k\|^2 - \|y_{i_k}^{k-1}-y_{i_k}^k\|^2].\end{aligned} \tag{A.0.44}$$

Since $\bar{\Delta}_k$ defined above shares a similar structure with $\tilde{\Delta}_k$ in (A.0.36), we can follow a similar procedure as in Proposition A.0.6 to obtain a bound for $\mathbb{E}\{Q(\bar{\mathbf{z}}^N,\mathbf{z})\}$. Note that the only difference between (A.0.44) and (A.0.36) exists in the coefficient of the term $\mathbf{V}(\mathbf{x}^{k-1},\mathbf{x})$

and $\mathbf{V}(\mathbf{x}^k, \mathbf{x})$. Hence, by using condition (A.0.39) in place of (A.0.27), we obtain

$$\mathbb{E}\left\{Q(\bar{\mathbf{z}}^N, \mathbf{z})\right\} \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x})) + m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right)\mathbf{V}(\mathbf{x}^0, \mathbf{x}) + \textstyle\sum_{k=1}^{N} \tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)(\eta_k+\mu)}$$
$$+ \mathbb{E}\left\{\langle \hat{\theta}_0 \mathbf{L}\mathbf{x}^0 + \hat{\theta}_N \mathbf{L}(\underline{\hat{\mathbf{x}}}^N - \mathbf{x}^{N-1}) + m\hat{\theta}_1\tau_1\mathbf{y}^N, \mathbf{y}\rangle\right\}.$$

Our result in (A.0.40) immediately follows. Following the same procedure as we obtain (A.0.38), for any saddle point $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ of (4.2.1), we have

$$\tfrac{\hat{\theta}_N\eta_N}{4}\mathbb{E}\|\underline{\hat{\mathbf{x}}}^{N-1} - \mathbf{x}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right)\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$$
$$+ \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 + \mathbb{E}\left\{\tfrac{\hat{\theta}_N\|\mathbf{L}\|^2}{2m\tau_N}\|\underline{\hat{\mathbf{x}}}^{N-1} - \mathbf{x}^N\|^2\right\} + \textstyle\sum_{k=1}^{N}\tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)(\eta_k+\mu)},$$
$$\tfrac{m\hat{\theta}_N\tau_N}{2}\mathbb{E}\|\mathbf{y}^* - \mathbf{y}^N\|^2 \leq \hat{\theta}_0(F(\mathbf{x}^0) - F(\mathbf{x}^*) + \langle \mathbf{L}\mathbf{x}^0, \mathbf{y}^*\rangle) + m\hat{\theta}_1\left(\tfrac{4(\mathcal{C}+L)}{T_1(T_1+1)} + \eta_1\right)\mathbf{V}(\mathbf{x}^0, \mathbf{x}^*)$$
$$+ \tfrac{m\hat{\theta}_1\tau_1}{2}\|\mathbf{y}^*\|^2 + \mathbb{E}\{\tfrac{m\hat{\theta}_N\|\mathbf{L}\|^2}{\eta_N}\|\mathbf{y}^* - \mathbf{y}^N\|^2\} + \textstyle\sum_{k=1}^{N}\tfrac{8m(M^2+\sigma^2)\hat{\theta}_k}{(T_k+1)(\eta_k+\mu)},$$

from which the result in (A.0.41) follows. ■

REFERENCES

[1] T. Strohmer and R. Vershynin, "A randomized kaczmarz algorithm with exponential convergence," *Journal of Fourier Analysis and Applications*, vol. 15, no. 2, pp. 262–278, 2009.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, "Communication-efficient learning of deep networks from decentralized data," *ArXiv preprint arXiv:1602.05629*, 2016.

[3] A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *Siopt*, vol. 19, pp. 1574–1609, 2009.

[4] A. Juditsky and A. S. Nemirovski, *First-Order Methods for Nonsmooth Convex Large-Scale Optimization, I: General Purpose Methods*, ser. in Optimization for Machine Learning, Eds: S. Sra, S. Nowozin and S.J. Wright. MIT press, 2011.

[5] Y. E. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady AN SSSR*, vol. 269, pp. 543–547, 1983.

[6] ——, *Introductory lectures on convex optimization: A basic course*. Massachusetts: Kluwer Academic Publishers, 2004.

[7] ——, "Gradient methods for minimizing composite objective functions," *Mathematical Programming., Series B*, vol. 140, pp. 125–161, 2013.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[9] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," University of Washington, Seattle, Manuscript, May 2008.

[10] G. Lan, "An optimal method for stochastic composite optimization," *Mathematical Programming*, vol. 133(1), pp. 365–397, 2012.

[11] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: A generic algorithmic framework," *SIAM Journal on Optimization*, vol. 22, pp. 1469–1492, 2012.

[12] ——, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: Shrinking procedures and optimal algorithms," *Siopt*, vol. 23, pp. 2061–2089, 2013.

[13] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey.," in *Optimization for Machine Learning*, S. N. S. Sra and S. J. Wright, Eds., Extended version: LIDS report LIDS-P2848, MIT, 2010, MIT Press, 2012, pp. 85–119.

[14] D. Blatt, A. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, 2007.

[15] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances of Neural Information Processing Systems (NIPS)*, vol. 26, pp. 315–323, 2013.

[16] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[17] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," *Advances of Neural Information Processing Systems (NIPS)*, vol. 27, 2014.

[18] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, vol. 162, no. 1-2, pp. 83–112, 2017.

[19] G. Lan and Y. Zhou, "An optimal randomized incremental gradient method," *Mathematical programming*, pp. 1–49, 2017.

[20] Z. Allen-Zhu, "Katyusha: The first direct acceleration of stochastic gradient methods," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, ACM, 2017, pp. 1200–1205.

[21] E. Hazan and H. Luo, "Variance-reduced and projection-free stochastic optimization," *CoRR, abs/1602.02101*, vol. 2, 2016.

[22] H. Lin, J. Mairal, and Z. Harchaoui, "A universal catalyst for first-order optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 3384–3392.

[23] A. S. Nemirovski and D. Yudin, *Problem complexity and method efficiency in optimization*, ser. Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.

[24] G. Lan, "Efficient methods for stochastic composite optimization," Georgia Institute of Technology, Manuscript, 2008.

[25] Y. Zhang and L. Xiao, "Stochastic primal-dual coordinate method for regularized empirical risk minimization," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 353–361.

[26] C. Dang and G. Lan, "Randomized first-order methods for saddle point optimization," Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, Manuscript, September 2014.

[27] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Mathematical Programming*, 2015, to appear.

[28] ——, "Stochastic dual coordinate ascent methods for regularized loss," *Journal of Machine Learning Research*, vol. 14(1), 567599, 2013.

[29] Q. Lin, Z. Lu, and L. Xiao, "An accelerated proximal coordinate gradient method," in *Advances in Neural Information Processing Systems*, 2014, pp. 3059–3067.

[30] Y. E. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep., 2010.

[31] O. Fercoq and P. Richtárik, "Smooth minimization of nonsmooth functions with parallel coordinate descent methods," *ArXiv e-prints*, 2013. arXiv: 1309.5885.

[32] A. Agarwal and L. Bottou, "A lower bound for the optimization of finite sums," *ArXiv e-prints*, 2014. arXiv: 1410.0723.

[33] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams, "Variance reduced stochastic gradient descent with neighbors," in *Advances in Neural Information Processing Systems*, 2015, pp. 2305–2313.

[34] F. Pedregosa, R. Leblond, and S. Lacoste-Julien, "Breaking the nonsmooth barrier: A scalable parallel method for composite optimization," in *Advances in Neural Information Processing Systems*, 2017, pp. 55–64.

[35] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *Journal of Machine Learning Research*, vol. 13, no. Jan, pp. 165–202, 2012.

[36] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An asynchronous mini-batch algorithm for regularized stochastic optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3740–3754, 2016.

[37] M. Rabbat and R. D. Nowak, "Distributed optimization in sensor networks," in *IPSN*, 2004, pp. 20–27.

[38] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988 –1001, 2003.

[39] S. S. Ram, V. V. Veeravalli, and A. Nedić, "Distributed non-autonomous power control through distributed convex optimization," in *IEEE INFOCOM*, 2009, pp. 3001–3005.

[40] J. W. Durham, A. Franchi, and F. Bullo, "Distributed pursuit-evasion without mapping or global localization via local frontiers," *Autonomous Robots*, vol. 32, no. 1, pp. 81–95, 2012.

[41] R. A. Hom and C. R. Johnson, "Topics in matrix analysis," *Cambridge UP, New York*, 1991.

[42] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," PhD thesis, Massachusetts Inst. Technol., Cambridge, MA, 1984.

[43] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803 –812, 1986.

[44] A. Nedić, D. P. Bertsekas, and V. S. Borkar, "Distributed asynchronous incremental subgradient methods," *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, pp. 311–407, 2001.

[45] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM J. on Optimization*, vol. 20, no. 2, pp. 691–717, 2009.

[46] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Mathematical Programming*, vol. 129, 163–195, 2011.

[47] M. Wang and D. P. Bertsekas, "Incremental constraint projection-proximal methods for nonsmooth convex optimization," Laboratory for Information and Decision Systems, Tech. Rep. LIDS-P-2907, 2013.

[48] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo, "On the convergence rate of incremental aggregated gradient algorithms," http://arxiv.org/abs/1506.02081, 2015.

[49] D. P. Bertsekas, "Incremental aggregated proximal and augmented lagrangian algorithms," Laboratory for Information and Decision Systems, Tech. Rep. LIDS-P-3176, 2015.

[50] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[51] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Automat. Contr.*, vol. 57, no. 3, pp. 592–606, 2012.

[52] M. Zhu and S. Martinez, "On distributed convex optimization under inequality and equality constraints," *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 151–164, 2012.

[53] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Automat. Contr.*, vol. 56, no. 6, pp. 1337–1351, 2011.

[54] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[55] K. Tsianos, S. Lawlor, and M. Rabbat, "Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning," in *Proceedings of the 50th Allerton Conference on Communication, Control, and Computing*, 2012.

[56] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, 944–966, 2015.

[57] ——, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, 6013–6023, 2015.

[58] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," http://arxiv.org/abs/1605.07112, 2016.

[59] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," http://arxiv.org/abs/1607.03218, 2016.

[60] H. Terelius, U. Topcu, and R. Murray, "Decentralized multi-agent optimization via dual decomposition," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 245–11 251, 2011.

[61] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[62] W. Shi, Q. Ling, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[63] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed admm over networks," http://arxiv.org/abs/1601.00194, 2016.

[64] E. Wei and A. Ozdaglar, "On the $O(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," http://arxiv.org/pdf/1307.8254, 2013.

[65] N. S. Aybat, Z. Wang, T. Lin, and S. Ma, "Distributed linearized alternating direction method of multipliers for composite convex consensus optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 1, pp. 5–20, 2018.

[66] N. He, A. Juditsky, and A. Nemirovski, "Mirror prox algorithm for multi-term composite minimization and semi-separable problems," *Journal of Computational Optimization and Applications*, vol. 103, pp. 127–152, 2015.

[67] T. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," http://arxiv.org/abs/1402.6065, 2014.

[68] T. Chang and M. Hong, "Stochastic proximal gradient consensus over random networks," http://arxiv.org/abs/1511.08905, 2015.

[69] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "Dqm: Decentralized quadratically approximated alternating direction method of multipliers," http://arxiv.org/abs/1508.02073, 2015.

[70] ——, "A decentralized second-order method with exact linear convergence rate for consensus optimization," http://arxiv.org/abs/1602.00596, 2016.

[71] D. Jakovetic, J. Xavier, and J. Moura, "Fast distributed gradient methods," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1131–1145, 2014.

[72] A. Chen and A. Ozdaglar, "A fast distributed proximal gradient method," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, 2012, pp. 601–608.

[73] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," *Journal of Optimization Theory and Applications*, vol. 147, pp. 516–545, 3 2010.

[74] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015, pp. 517–520.

[75] C. Xi, Q. Wu, and U. A. Khan, "Distributed mirror descent over directed graphs," http://arxiv.org/abs/1412.5526, 2014.

[76] A. Simonetto, L. Kester, and G. Leus, "Distributed time-varying stochastic optimization and utility-based communication," http://arxiv.org/abs/1408.5294, 2014.

[77] K. Tsianos and M. Rabbat, "Consensus-based distributed online prediction and optimization," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 807–810.

[78] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[79] S. Lee and A. Nedić, "Gossip-based random projection algorithm," in *Proceedings of the 46th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2012.

[80] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 434–448, 2018.

[81] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," http://arxiv.org/pdf/1303 2013.

[82] G. Zhang and R. Heusdens, "Bi-alternating direction method of multipliers over graphs," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, IEEE, 2015, pp. 3571–3575.

[83] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[84] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," in *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, IEEE, 2016, pp. 992–996.

[85] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[86] L. Bregman, "The relaxation method of finding the common point convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Phys.*, vol. 7, pp. 200–217, 1967.

[87] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *Siopt*, vol. 16, pp. 697–725, 2006.

[88] H. Bauschke, J. Borwein, and P. Combettes, "Bregman monotone optimization algorithms," *SIAM Journal on Controal and Optimization*, vol. 42, pp. 596–636, 2003.

[89] K. Kiwiel, "Proximal minimization methods with generalized bregman functions," *SIAM Journal on Controal and Optimization*, vol. 35, pp. 1142–1168, 1997.

[90] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[91] W. L. Winston and J. B. Goldberg, *Operations research: Applications and algorithms*. Duxbury press Belmont, CA, 2004, vol. 3.

[92] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Mathematical Programming*, vol. 156, no. 1-2, pp. 59–99, 2016.

[93] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vision*, vol. 40, pp. 120–145, 2011.

[94] ——, "On the ergodic convergence rates of a first-order primal-dual algorithm," Oct. 30, 2014.

[95] G. Lan, A. S. Nemirovski, and A. Shapiro, "Validation analysis of mirror descent stochastic approximation method," *Mathematical Programming*, vol. 134, pp. 425–458, 2012.

[96] Z. Allen-Zhu and E. Hazan, "Optimal black-box reductions between optimization objectives," *ArXiv preprint arXiv:1603.05642*, 2016.

[97] Y. E. Nesterov, "Smooth minimization of nonsmooth functions," *Mathematical Programming*, vol. 103, pp. 127–152, 2005.

[98] Y. Nesterov, "Unconstrained convex minimization in relative scale," *Mathematics of Operations Research*, vol. 34, no. 1, pp. 180–193, 2009.

[99] G. Lan, Z. Lu, and R. D. C. Monteiro, "Primal-dual first-order methods with $\mathcal{O}(1/\epsilon)$ iteration-complexity for cone programming," *Mprog*, vol. 126, pp. 1–29, 2011.

[100] Y. Censor and A. Lent, "An iterative row-action method for interval convex programming," *Journal of Optimization theory and Applications*, vol. 34, no. 3, pp. 321–353, 1981.

[101] R. D. C. Monteiro and B. F. Svaiter, "On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2755–2787, 2010.

[102] ——, "Complexity of variants of tseng's modified f-b splitting and korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1688–1720, 2011.

[103] ——, "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 475–507, 2013.

[104] Y. Ouyang, Y. Chen, G. Lan, and E. P. Jr., "An accelerated linearized alternating direction method of multipliers," *SIAM Journal on Imaging Sciences*, vol. 8, no. 1, pp. 644–681, 2015.

[105] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in linear and non-linear programming*, ser. Stanford Mathematical Studies in the Social Sciences. Stanford University Press, 1958.

[106] A. S. Nemirovski, "Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *Siopt*, vol. 15, pp. 229–251, 2005.

[107] R. Monteiro and B. Svaiter, "On the complexity of the hybrid proximal projection method for the iterates and the ergodic mean," *Siopt*, vol. 20, pp. 2755–2787, 2010.

[108] B. He and X. Yuan, "On the o(1/n) convergence rate of the douglas-rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[109] Y. Chen, G. Lan, and Y. Ouyang, "Optimal primal-dual methods for a class of saddle point problems," *SIAM Journal on Optimization*, vol. 24(4), pp. 1779–1814, 2014.

[110] N. S. Aybat and E. Y. Hamedani, "A primal-dual method for conic constrained distributed optimization problems," in *Advances in Neural Information Processing Systems*, 2016, pp. 5049–5057.

[111] G. Lan, "Gradient sliding for composite optimization," *Mathematical Programming*, vol. 159, no. 1, pp. 201–235, 2016.

[112] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization," Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, Manuscript, July 2010.

[113] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, "1-norm support vector machines," in *Advances in neural information processing systems*, 2004, pp. 49–56.

[114] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines.," in *ICML*, vol. 98, 1998, pp. 82–90.

[115] Q. Deng, G. Lan, and A. Rangarajan, "Randomized block subgradient methods for convex nonsmooth and stochastic optimization," *ArXiv preprint arXiv:1509.04609*, 2015.

[116] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: Shrinking procedures and optimal algorithms," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2061–2089, 2013.

VITA

Yi Zhou was born on October 14, 1992 in the city of Ganzhou, Jiangxi, China. After obtaining her B.S. degree in Information and Computational Science from Sun Yat-sen University in June 2013, she was admitted to the doctoral program of Industrial Engineering in the Department of Industrial and Systems Engineering at the University of Florida. In January 2016, she moved with her Ph.D. adviser and transferred to Georgia Institute of Technology to continue pursuing her Ph.D. degree in the H. Milton Stewart School of Industrial and Systems Engineering. She completed her Ph.D degree in early August 2018. She will join IBM Research at Almaden as a research staff member in August 2018.